



Universidad
Carlos III de Madrid

Departamento de Ingeniería Térmica y de Fluidos

PROYECTO FIN DE CARRERA

Ingeniería Industrial

**ARTIFICIAL NEURAL NETWORK (ANN)
COMO MÉTODO DE SIMULACIÓN DE
LECHOS FLUIDOS BIDIMENSIONALES
BURBUJEANTES**

Autor: Ricardo Jesús Boada Corredera

Tutor: Sergio Sanchez Delgado

Leganés, Octubre de 2013

Título: ARTIFICIAL NEURAL NETWORK (ANN) COMO MÉTODO DE
SIMULACIÓN DE LECHOS FLUIDOS BIDIMENSIONALES BURBUJEANTES

Autor: RICARDO BOADA CORREDERA

Director: SERGIO SANCHEZ DELGADO

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ____ de _____
de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Este proyecto va dedicado a todas las personas que me han acompañado y ayudado durante estos años de universidad que llegan a su fin.

En especial, se lo dedico a mis padres, a mi hermana Laura, a Rares y a Paulina.

Índice general

1 INTRODUCCIÓN Y OBJETIVOS	8
1.1 Introducción	9
1.2 Objetivos	10
2 LECHOS FLUIDIZADOS	11
2.1 Definición.....	12
2.2 Principios operativos.....	12
2.3 Caracterización de la Fluidización	15
2.4 Pérdida de carga y mínima fluidización.....	17
2.5 Diagrama de caída de presión frente a la velocidad.....	18
2.6 Formación y crecimiento de las burbujas.....	19
2.7 Algunas aplicaciones de los lechos fluidizados	21
3 REDES NEURONALES ARTIFICIALES	22
3.1 Introducción y principios de funcionamiento	23
3.2 Inspiración biológica	24
3.3 Tipos de redes neuronales artificiales	25
3.4 Fundamentos generales del proceso de aprendizaje.....	25
3.5 Componentes de la neurona artificial.....	32
3.6 La ADALINA (ADaptive LInear NEuron).....	33
3.7 El Perceptrón Multicapa (MLP).....	36
3.8 Aplicaciones de las Redes neuronales.....	39
4 SELECCIÓN DE UNA RED NEURONAL PARA SIMULAR DIÁMETRO Y VELOCIDAD EN UN LECHO EXPERIMENTAL.....	40
4.1 Obtención de los datos experimentales	41
4.2 Análisis y preproceso de los datos experimentales	43
4.3 Variables para el diseño de la red y proceso de selección	50
4.4 Resultado de las simulaciones.....	52
4.5 Conclusiones y resumen del capítulo	62
TRABAJO FUTURO	63
REFERENCIAS BIBLIOGRÁFICAS	64

Índice de figuras

Fig 2.1. Comportamiento del lecho similar al de un líquido [KL].....	13
Fig 2.2. Estados de operación en función del Caudal. Elaboración Propia	14
Fig 2.3. Diferentes estados de operación de un lecho fluidizado [KL].....	16
Fig 2.4. Clasificación Geldart de las partículas [KL].....	17
Fig 2.5. La fuerza de fricción compensa el peso de las partículas [RN].....	18
Fig 2.6. Pérdida de presión en un lecho según la velocidad de entrada del fluido [KL] ..	20
Fig 2.7. Comportamiento de las burbujas en la región de entrada al lecho según el tipo de distribuidor (a) placa porosa (b) plato perforado (c) campana con orificios (d) campana con tejado [KL]	20
Fig 2.8. Proceso de separación y posterior re-coalescencia de una burbuja	21
Fig 3.1. Caja Entrada-Salida [CS].....	24
Fig 3.2. Estructura básica de una neurona biológica [NEU].....	25
Fig 3.3. Función con ruido aleatorio (rojo) frente a función sin ruido (negro).....	27
Fig 3.4. Resultado red sin entrenar (verde) y datos sin ruido (negro).....	28
Fig 3.5. Resultado red entrenada (verde) frente a los datos sin ruido (negro).	29
Fig 3.6. Resultado red entrenada (verde) frente a función sin ruido (negro), y los datos con ruido representados con círculos (azules). Elaboración Propia.....	30
Fig 3.7. Esquema de la división del conjunto de datos para mejorar la capacidad de generalización. Elaboración Propia.....	31
Fig 3.8. Resultados de entrenamiento con 5 neuronas y Validation Stop.....	32
Fig 3.9. Resultados de entrenamiento con 30 neuronas sin Validation Stop	32
Fig 3.10 Modelo de neurona estándar [MS].....	33
Fig 3.11. Funciones de activación [Izq]	34
Fig 3.12. Estructura de la Adalina [MAT]	35
Fig 3.13. Ilustración del método de descenso por gradiente [MS].....	36
Fig 3.14. Estructura general de una red de Perceptrón Multicapa. Elaboración Propia....	38
Fig 3.15. Esquema de un perceptrón multicapa de una capa oculta [MS]	38
Fig 4.1. (a) Lecho experimental de la UC3M [ISE] (b) Fotografía de burbujas en el lecho [Izq].....	42
Fig 4.2. Sección transversal del lecho experimental de la UC3M [Izq]	44
Fig 4.3. Diagrama que muestra la dispersión de los datos. Elaboración Propia	45
Fig 4.4. Diámetro de la burbuja en función de y (h=30 cm, q=94 l/min)	46
Fig 4.5. Diámetro de la burbuja en función de x (h=30 cm, q=94 l/min)	46
Fig 4.6. Componente vertical de velocidad de la burbuja en función de y (h=30 cm, q=94 l/min).....	48
Fig 4.7. BoxPlot para el diámetro de burbuja en función de la altura.....	49
Fig 4.8. Número de burbujas observadas en función de la posición vertical en el lecho. Elaboración Propia	49
Fig 4.9. Gráfico explicativo del proceso de filtrado de datos por intervalos de confianza. Elaboración Propia	50
Fig 4.10. Resultados del entrenamiento con datos en bruto. Elaboración Propia.....	51
Fig 4.11. Estructura de la red con 1 capa oculta. Elaboración Propia.....	53
Fig 4.12. Estructura de la red con 2 capas ocultas. Elaboración Propia.....	58
Fig 4.13. Estructura de la red con 1 capa oculta y 2 neuronas de salida. Elaboración Propia	61

Índice de tablas

Tabla 4.1. Combinaciones de datos utilizados en función de la velocidad y altura de lecho fijo	42
Tabla 4.10. Resumen de los diferentes entrenamientos de una estructura de red con resultados atípicos	60
Tabla 4.11. Resultados predicción diámetro y velocidad para Conjunto de Test. Red 2 capas ocultas y 2 neuronas en la capa de salida	61
Tabla 4.2. Datos de partida, en la forma “BB_Velo2_altura lecho fijo(cm)_caudal (l/min)”	42
Tabla 4.3. Resultados predicción diámetro para Conjunto de Test. Red 1 capa oculta	53
Tabla 4.4. Salida del programa informático Minitab para un contraste de hipótesis.....	54
Tabla 4.5. Resultados predicción velocidad para Conjunto de Test. Red 1 capa oculta...	55
Tabla 4.6. Salida del programa informático Minitab para un contraste de hipótesis.....	56
Tabla 4.7. Resultados predicción diámetro para Conjunto de Test. Red 2 capas ocultas .	57
Tabla 4.8. Resumen de los diferentes entrenamientos de una estructura de red con resultados atípicos	58
Tabla 4.9. Resultados predicción diámetro para Conjunto de Test. Red 2 capas ocultas .	59

1 Introducción y objetivos

1.1 Introducción

Un *lecho fluidizado* es un recipiente que contiene un conjunto de partículas sólidas, a través del cual se hace pasar un fluido que se dirige de abajo a arriba, y hace que las partículas sólidas queden suspendidas por acción del rozamiento que compensa el peso de las mismas, que flotan inmersas en la corriente fluida. El conjunto de partículas sólidas comienza entonces a moverse y comportarse en muchos aspectos como si de un fluido se tratase. Esto crea una combinación de propiedades que resulta muy interesante para un amplio abanico de procesos industriales, sobre todo en operaciones de contacto sólido-fluido en las que, por ejemplo, se requieran simultáneamente buenas propiedades de transporte (por comportarse como si fuera un fluido en movimiento), y buenas capacidades de almacenamiento térmico (¡las partículas no dejan de ser sólidas!).

Si el fluido de trabajo es un gas, pongamos por caso aire, y las partículas sólidas son lo suficientemente grandes (como es el caso de la arena), al incrementarse el flujo de aire, comenzarán a formarse burbujas: regiones en las que apenas encontramos material sólido. Diremos en este caso, uno de los más utilizados y el que nos ocupa, que tenemos un lecho fluidizado burbujeante.

El aire atraviesa entonces el reactor en forma de emulsión a través del espacio vacío entre los sólidos, o en forma de burbujas que van creciendo, coalesciendo y separándose. Estas burbujas que cruzan el lecho provocan una fuerte agitación y mezcla de los sólidos, tanto en la componente radial (para lechos bidimensionales, transversal), como en la longitudinal, gobernando en buena medida los procesos que ocurren en su interior. Todo esto implica un problema de muy difícil modelización analítica: partículas fluidas que entran en el lecho al mismo tiempo y lo abandonan en diferente instante, los sólidos que ascienden empujados por las burbujas y luego caen otra vez en un movimiento con un alto grado de aleatoriedad, etc. Es por eso que para diseñar y caracterizar estos lechos generalmente se hace necesario recurrir a procedimientos experimentales y simulaciones numéricas con modelos complejos de CFD (Computational Fluid Dynamics), destaquemos entre ellos los métodos DBM (Discrete Bubble Model).

Durante el presente proyecto, utilizaremos un método computacional llamado *Redes Neuronales Artificiales (Artificial Neural Networks)* para simular ciertas variables de un lecho fluidizado a partir de datos experimentales, de manera que sirva para reducir el número de experimentos necesarios para caracterizar a un lecho y comprender su funcionamiento, y a modo de validación de los modelos de CFD antes mencionados con el consiguiente ahorro de coste computacional. Esta herramienta que utilizaremos, las redes neuronales, surgieron en los años 40 como un nuevo paradigma de procesamiento automático de la información, que intenta reproducir en una máquina el proceso de tratamiento y análisis de la información que realizamos los sistemas biológicos, imitando su estructura.

Una red neuronal actúa como una función matemática, realizando una correspondencia entre elementos de dos conjuntos. Transforma unos datos de entrada en otros de salida, pero sin utilizar una expresión analítica. La información fluye de manera dinámica desde la entrada a la salida, pasando por un conjunto de elementos de procesamiento interconectados, llamados neuronas, que realizan una serie de operaciones

y permiten que continúe el flujo de la información, enviándosela a las siguientes neuronas hasta alcanzar el conjunto de salida.

En la actualidad, los métodos de computación neuronal están cobrando cada vez más importancia debido a su potencia para resolver diversos problemas reales en los que otras herramientas clásicas ofrecen peores resultados o no es posible obtener una solución analítica, ofreciendo una muy buena relación general eficiencia/coste.

1.2 Objetivos

El presente proyecto pretende ajustar y simular ciertas variables de interés (diámetro y velocidad de burbuja) de un lecho fluidizado burbujeante bidimensional a través de unos datos experimentales en un lecho real obtenidos por el grupo de investigación ISE (Ingeniería de Sistemas Energéticos) de la Universidad Carlos III de Madrid.

Para ello se diseñará una red neuronal artificial en el software comercial Matlab R2009a, y se optimizará su funcionamiento mediante la selección adecuada de la topología de red y a través de un preprocesado estadístico de los datos que se usarán para el aprendizaje de la red.

El resultado de este proyecto servirá en el futuro para reducir el número de experimentos necesarios para la caracterización del lecho fluidizado, y como modelo de validación de simulaciones más complejas elaboradas con métodos de CFD. Así mismo, la metodología de análisis de los datos y diseño de la red neuronal óptima, es extensible de manera inmediata para caracterizar cualquier tipo de lecho fluidizado, y en general puede resultar aplicable a un número mayor de problemas en los que se pretenda simular mediante métodos neuronales los valores promedio de datos con elevada dispersión.

2 Lechos fluidizados

2.1 Definición

Un lecho fluidizado es un conjunto de partículas sólidas, confinadas en una vasija o reactor, a través de las cuales se hace pasar una corriente vertical de fluido provocando que las partículas sólidas, al encontrarse en suspensión en la corriente fluida, se comporten en muchos aspectos como si de un fluido se tratase.

El conjunto de partículas sólidas, moviéndose a través del lecho, adquiere entonces muchas propiedades que se asemejan a las que presentan los líquidos (mantienen la superficie libre horizontal al inclinarse, los objetos ligeros que se sumerjan flotarán,...), tal y como muestra la figura inferior. El proceso por el cual se produce este fenómeno se conoce como **fluidización**.

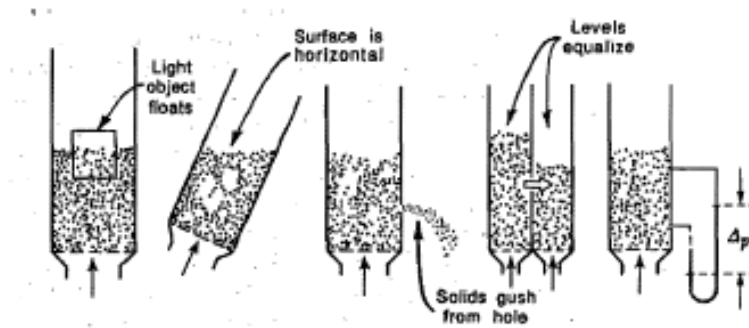


Fig 2.1. Comportamiento del lecho similar al de un líquido [KL]

2.2 Principios operativos

Estudiaremos ahora los distintos estados de operación que pueden presentar estos sistemas según vamos incrementando la velocidad de entrada del fluido.

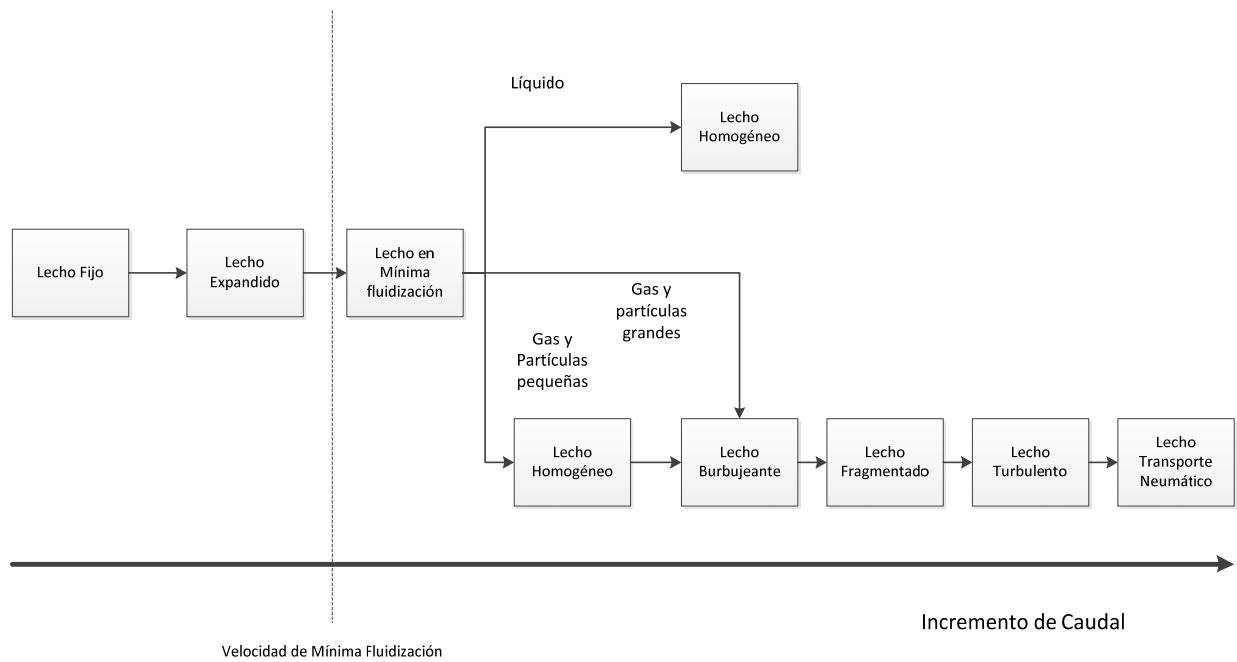


Fig 2.2. Estados de operación en función del Caudal. Elaboración Propia

- Lecho Fijo: Si hacemos circular un caudal pequeño de fluido a través del lecho, éste pasará por los espacios vacíos existentes entre las partículas sólidas, que permanecerán en reposo.
- Lecho Expandido: Si vamos aumentando progresivamente el caudal de fluido, las partículas sólidas comenzarán a moverse ligeramente por efecto del rozamiento partícula sólida/fluido, y se experimentarán vibraciones en ciertas zonas, especialmente en la superficie.
- Mínima fluidización: Si continuamos incrementando el caudal, llegará un punto en el cuál la fuerza de rozamiento ocasionada por el flujo del fluido a través de las partículas compensará el peso de las mismas, alcanzándose un estado en el cual las partículas sólidas quedan suspendidas en el flujo y la componente vertical de los esfuerzos compresivos entre partículas adyacentes desaparece.

A partir de este momento, si se aumenta aún más la velocidad de entrada del fluido, las partículas sólidas se mantendrán en suspensión sobre la corriente, y diremos entonces que **el lecho está fluidizado**. Es decir, el estado de mínima fluidización es un transitorio (difícilmente observable) entre un lecho expandido y un lecho fluidizado.

- Lecho Homogéneamente fluidizado: Para el caso que tengamos un sistema líquido-sólido, tendremos que un incremento en el caudal de entrada de líquido por encima de las condiciones de mínima fluidización, resulta en una expansión progresiva y continúa del lecho.
- Lecho Burbujeante: Para sistemas en los que la corriente fluida está en estado gaseoso, al sobrepasar las condiciones de mínima fluidización, comienzan a formarse burbujas (regiones con muy baja presencia de partículas sólidas), que aparecen antes cuanto mayor sea el tamaño de las partículas (Si el tamaño de las mismas es pequeño, primeramente aparece un estado homogéneo similar al de los lechos con líquido, que acaba desembocando en lecho burbujeante).
- Lecho Fragmentado (Slugging): Para sistemas gas-sólido, un aumento del caudal más allá del que origina un lecho burbujeante, puede resultar en la coalescencia masiva de burbujas, en forma de adherencia parcial a las paredes (Slug axial), o transversalmente (Slug Plano). En este caso, estas zonas actúan como tapones de gas, alternados con zonas de sólidos fluidizados que son transportados hacia arriba, y que posteriormente se deshacen precipitándose después los sólidos hacia abajo en forma de lluvia, en un proceso inestable y repetitivo. Se presenta especialmente en sistemas esbeltos con altos caudales, que permiten que las burbujas crezcan rápidamente mientras ascienden coalesciendo hasta alcanzar el diámetro de la vasija.
- Lechos Turbulentos o rápidos: Finalmente, cuando las partículas son fluidizadas a un caudal lo suficientemente elevado, la superficie superior desaparece y los sólidos comienzan a salir fuera de la vasija. En lugar de burbujas observamos regiones de sólidos agrupados que se alternan con regiones de vacías ocupadas por gas, todo ello inmerso en un movimiento altamente aleatorio.
- Lechos de Transporte neumático: Por encima del estado de fluidización turbulenta, tenemos que los sólidos salen fuera de la vasija junto a la fase gaseosa.

En los dos últimos casos, suele hacerse necesaria una recirculación de las partículas expulsadas en la mayoría de las aplicaciones para mantener un cierto grado de continuidad en las operaciones. Se dice entonces que tenemos Lechos Circulantes.

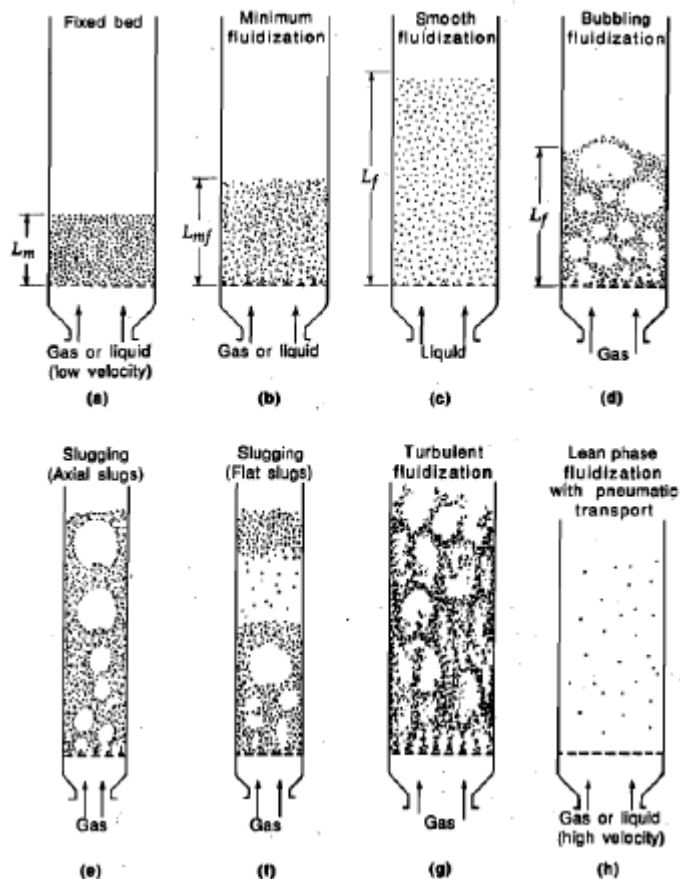


Fig 2.3. Diferentes estados de operación de un lecho fluidizado [KL]

2.3 Caracterización de la Fluidización

En esta sección se presentan las principales variables que influyen en el proceso de fluidización, y que nos permitirán a partir de correlaciones experimentales y relaciones teóricas predecir en cuál de los estados vistos en el apartado anterior operará nuestro lecho.

2.3.1 Caracterización de las partículas. Clasificación Geldart

Una variable de importancia en los lechos fluidos es el tamaño de las partículas que componen el lecho.

Para el caso en el que las mismas no son esféricas, definimos el diámetro esférico equivalente, d_{sph} como el diámetro de la esfera equivalente que tiene el mismo volumen que la partícula.

Consideramos también la esfericidad, ϕ_s como el cociente entre la superficie de la esfera que tenga el mismo volumen que la partícula y la superficie de la partícula misma.

A partir de estas magnitudes, definimos el diámetro efectivo de las partículas en el lecho como aquél tal que un lecho formado por partículas esféricas con ese diámetro tendría casi la misma resistencia por fricción y el mismo porcentaje de espacio vacío ε_m que el lecho original. Esto equivale a decir que el cociente $a = \text{Superficie/Volumen}$ debería de ser el mismo en ambos lechos:

$$a = \frac{S_{part}}{V_{part}} = \frac{S_{eff}}{V_{eff}} = \frac{6}{d_{eff}} \quad (ec\ 2.1)$$

$$a = \frac{S_{part}}{V_{part}} = \frac{S_{sph}/\phi}{V_{sph}} = \frac{\pi d_{sph}^2/\phi_s}{\pi d_{sph}^3/6} = \frac{6}{d_{sph}\phi_s} \quad (ec\ 2.2)$$

$$\text{Donde, juntando ambas relaciones: } d_{eff} = \phi_s \cdot d_{sph} \quad (ec\ 2.3)$$

Veamos ahora una clasificación de las partículas en función de sus propiedades para fluidización en condiciones ambiente. **Geldart** (1973) las clasificó en cuatro grupos en función del tamaño d_p y la densidad, que permiten predecir cómo se comportarán en el proceso de fluidización según pertenezcan a un grupo u otro.

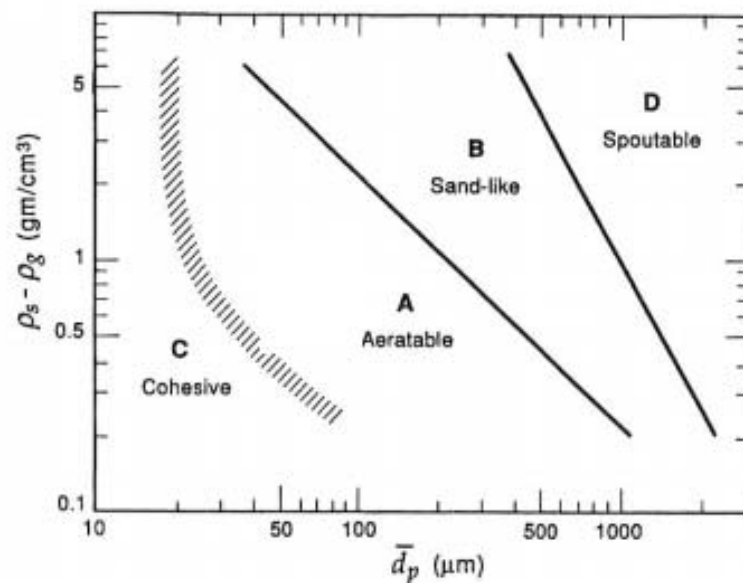


Fig 2.4. Clasificación Geldart de las partículas [KL]

2.4 Pérdida de carga y mínima fluidización

Como veíamos en el apartado 2.2, un lecho puede comportarse de diversas maneras en función del caudal. En este apartado veremos la dependencia de la caída de presión en función de la velocidad y demás parámetros de interés.

Mientras no se alcance la condición de mínima fluidización, el lecho será considerado como fijo y la pérdida de carga será debida a la fricción que ejercen las partículas sobre el fluido cuando éstas se encuentran quietas y el fluido circula entre ellas, aprovechando tan sólo los huecos disponibles. Ha sido correlacionada por Ergun [KL], según:

$$\frac{\Delta p_{fr}}{L_m} g_c = 150 \frac{(1-\varepsilon_m)^2}{\varepsilon_m^3} \frac{\mu u_0}{(\phi_s d_p)^2} + 1.75 \frac{1-\varepsilon_m}{\varepsilon_m^3} \frac{\rho_g u_0^2}{\phi_s d_p} \quad (ec\ 2.4)$$

A este término, deberemos de sumar, para el caso de líquidos, el debido al peso del fluido (que puede ser ignorada en el caso de gases), obteniendo:

$$\Delta p_{measured} = \Delta p_{fr} \pm \frac{\rho_g L_m}{g} \quad (ec\ 2.5) \quad , \quad \text{con } \Delta p \text{ siempre positiva.}$$

2.4.1 Punto de mínima fluidización

Una vez alcanzado el punto de mínima fluidización, se tendrá que cumplir que exista un estado de equilibrio en el que las partículas estén suspendidas, esto es, que la fuerza que el fluido ejerce sobre ellas por el rozamiento sea igual al peso de las partículas menos el empuje.

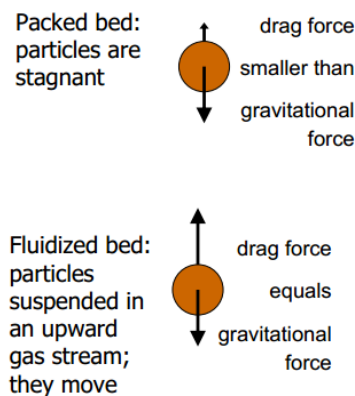


Fig 2.5. La fuerza de fricción compensa el peso de las partículas [RN]

Traduciendo este balance de fuerzas a ecuación, obtenemos para las condiciones de mínima fluidización:

$$\frac{\Delta p_b}{L_{mf}} = (1 - \varepsilon_{mf})(\rho_{sólido} - \rho_{fluido})g \quad (ec\ 2.6)$$

Dado que el cambio entre el estado de lecho fijo y mínima fluidización se trata de un proceso continuo, podemos combinar la ecuación anterior, con la que nos daba la pérdida de carga en el lecho fijo para obtener **la velocidad de mínima fluidización**.

Particularizando las variables de lecho fijo para condiciones de mínima fluidización nos queda una ecuación cuadrática en u_{mf} :

$$\frac{1.75}{\varepsilon_{mf}^3 \phi_s} \left(\frac{d_p u_{mf} \rho_{fluido}}{\mu} \right)^2 + \frac{150(1 - \varepsilon_{mf})}{\varepsilon_{mf}^3 \phi_s^2} \frac{d_p u_{mf} \rho_{fluido}}{\mu} = \frac{d_p^3 \rho_{fluido} (\rho_{sólido} - \rho_{fluido}) g}{\mu^2} \quad (ec\ 2.7)$$

2.5 Diagrama de caída de presión frente a la velocidad

Este diagrama constituye una buena indicación de la calidad de la fluidización. Para el caso de partículas no demasiado pequeñas y uniformes en forma y tamaño, podemos observar en la fig 2.6 como la curva muestra una tendencia creciente con el caudal de entrada, correspondiente a la zona de lecho fijo, donde la pérdida de carga es debida a la fricción con las partículas que se corresponde con la ec 2.5. El máximo marca el estado de mínima fluidización. A partir de ahí, se produce una variación del espacio vacío de ε_m a ε_{mf} , y consideramos que el lecho está fluidizado.

A partir de este punto, si seguimos incrementando el caudal, la caída de presión resulta prácticamente independiente de la velocidad de entrada de fluido, hasta el momento en el que se produzca un arrastre de las partículas a caudales muy elevados.

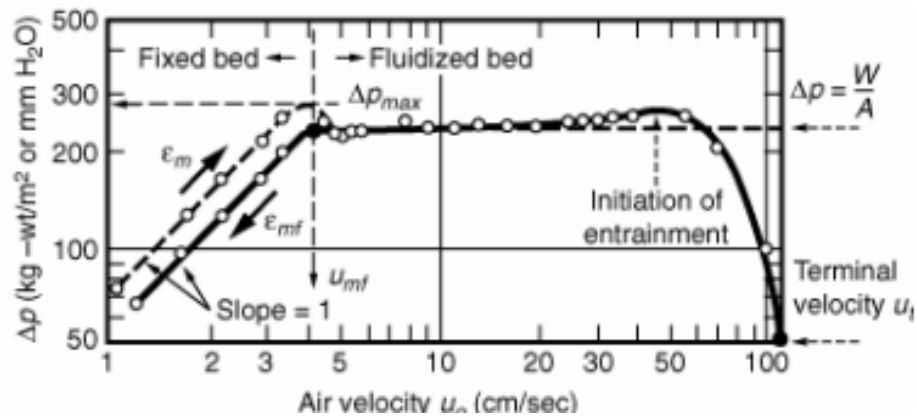


Fig 2.6. Pérdida de presión en un lecho según la velocidad de entrada del fluido [KL]

2.6 Formación y crecimiento de las burbujas

Primeramente, el fluido se introduce en el lecho a través de un **distribuidor**. Para su diseño, hay que intentar que la pérdida de carga sea lo suficientemente grande como para asegurar la distribución homogénea del fluido buscando un perfil plano de velocidades en la región de entrada, lo cuál por otra parte redunda en mayores costes de bombeo para instalaciones industriales.

Existen diferentes modelos de distribuidor, y la selección del mismo depende de la aplicación solicitada. En la siguiente figura podemos ver unas imágenes de cómo el tipo de distribuidor afecta al proceso de formación de burbujas en la entrada del lecho.

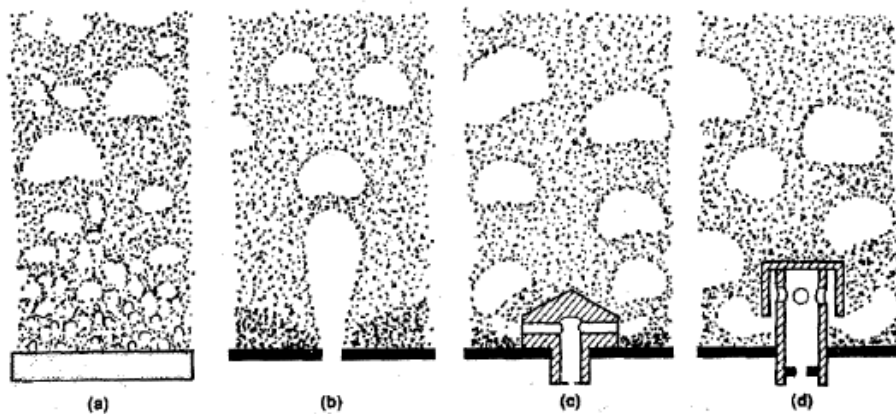


Fig 2.7. Comportamiento de las burbujas en la región de entrada al lecho según el tipo de distribuidor (a) placa porosa (b) plato perforado (c) campana con orificios (d) campana con tejado [KL]

Una vez nos alejemos de la zona de entrada, diversas correlaciones han sido obtenidas para intentar estimar el tamaño y velocidad de las burbujas en función de la altura. En general, son aplicables tan solo para lechos tridimensionales, no como en el que se utilizó en este proyecto que es bidimensional (de pequeño espesor). No obstante, y ya que es el objetivo del proyecto intentar ajustar estas variables, y que dichas correlaciones se utilizarán en una primera aproximación de las variables de estudio, se ha considerado necesario incluirlo en este capítulo. Para más detalles acudir a la obra de Kunni, D. [KL].

Para burbujas aisladas, y sin efecto pared, su velocidad de ascenso viene dada por (Ecuación idéntica a la de ascenso de una burbuja de gas inyectada en un líquido):

$$u_{br} = 0.711\sqrt{gd_b} \quad (ec\ 2.8)$$

En el caso de burbujas en un lecho fluidizado burbujeante, se añade un término que mide el exceso de aire respecto a las condiciones de mínima fluidización:

$$u_b = u_{br} + (u_0 - u_{mf}) \quad (ec\ 2.9)$$

Para el diámetro medio de las burbujas d_b , recurrimos a la siguiente ecuación (expresada en función del tamaño máximo esperable de una burbuja que se calcula a la derecha, y del diámetro del lecho d_t):

$$\frac{d_{b,max} - d_b}{d_{b,max} - d_{b0}} = \exp\left(-0.3z/d_t\right); \text{ con } d_{b,max} = 0.65\left[A_{transversal}(u_0 - u_{mf})\right]^{0.4} \text{ (unidades CGS) } (ec\ 2.10)$$

Otro proceso importante a tener en cuenta, es el de separación de burbujas. Tal y como se muestra en la figura, una burbuja puede comenzar a romperse en su parte superior por la formación de una cuña, que se propaga rápidamente hasta que la burbuja se ha dividido en dos. Poco después, y de manera casi inmediata si hay una diferencia importante entre el tamaño de ambas burbujas, puede producirse un fenómeno de coalescencia volviendo a unirse en una sola burbuja como en la figura, o puede ser que una de las burbujas consiga escaparse de la otra, dejando dos burbujas. El proceso de re-coalescencia es menos frecuente en lechos de partículas finas.

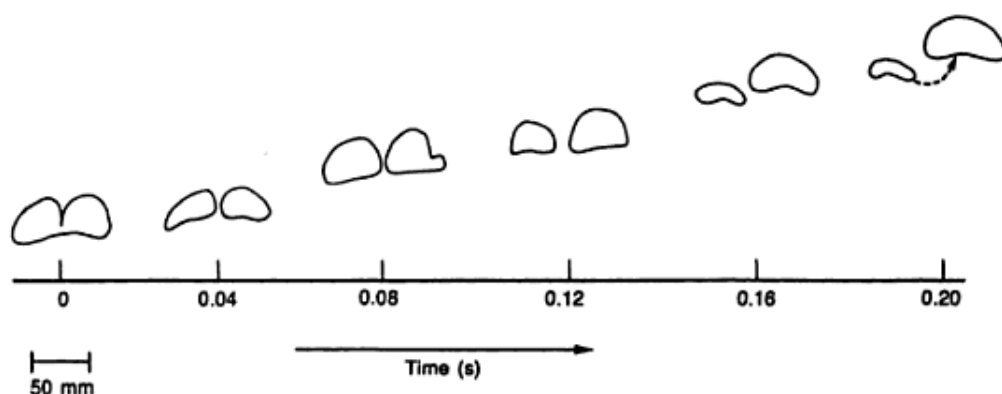


Fig 2.8. Proceso de separación y posterior re-coalescencia de una burbuja

2.7 Algunas aplicaciones de los lechos fluidizados

La combinación de propiedades que caracteriza a los lechos fluidizados los hace interesantes para un gran número de aplicaciones industriales, destaquemos entre ellas:

- Procesos físicos: Todos ellos aprovechan el excelente grado de mezclado que se produce.
 - Secado de sólidos tales como arena, polímeros y productos farmacéuticos.
 - Para operaciones de enfriamiento, debido a que el área de contacto sólido-fluido es muy elevada.
- Procesos químicos: Muchos de ellos utilizan el que el lecho se encuentre a temperatura prácticamente constante para ciertas reacciones en las que esto es un parámetro de diseño.
 - Citemos entre ellos el “Cracking” de hidrocarburos, en el que prácticamente siempre se hace necesario emplear un lecho fluidizado en alguna etapa del proceso.
 - Destaquemos también los procesos de combustión de biomasa, en los que también aprovechamos la facilidad de sustitución de las partículas sólidas debido al ensuciamiento.

3 Redes Neuronales Artificiales

3.1 Introducción y principios de funcionamiento

Una red neuronal es un sistema de procesamiento automático de información con una estructura inspirada en el sistema nervioso de los animales, que mediante un proceso de entrenamiento o aprendizaje, se prepara para generar la respuesta deseada ante un determinado estímulo.

Para tratar de entender este concepto, pensemos en una caja como la que se muestra en la siguiente figura:

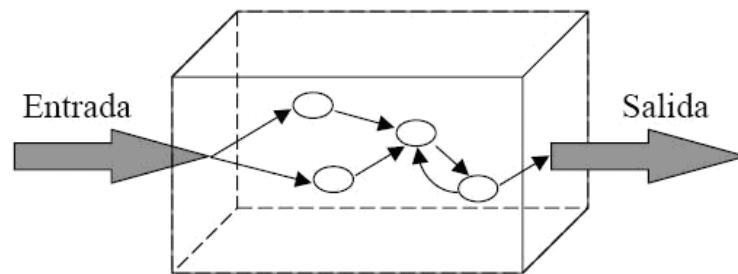


Fig 3.1. Caja Entrada-Salida [CS]

Dentro de la caja, tenemos un conjunto de nodos, llamados neuronas, que se envían información entre ellos, combinando y transformando los valores de entrada, hasta alcanzar la salida.

Una red neuronal, se comporta de manera análoga a una función matemática, produciendo salidas ante un conjunto de entradas. La principal ventaja es que esta función (pensada como relación entre elementos de dos conjuntos) no requiere de una expresión analítica, funciona de manera dinámica, propagando y modificando la información hasta alcanzar la salida.

De hecho, se puede demostrar que bajo ciertas condiciones de regularidad, algunos tipos de redes son aproximadores universales de funciones. [MS]

3.2 Inspiración biológica

El modelo de computación neuronal surge con una fuerte base biológica, con dos objetivos principales:

- Tratar de emular computacionalmente el proceso de razonamiento humano.
- Mejorar los conocimientos sobre el funcionamiento del sistema nervioso

Por ello considero obligado tratar, aunque sea a grandes rasgos, la estructura de las neuronas biológicas y su funcionamiento.

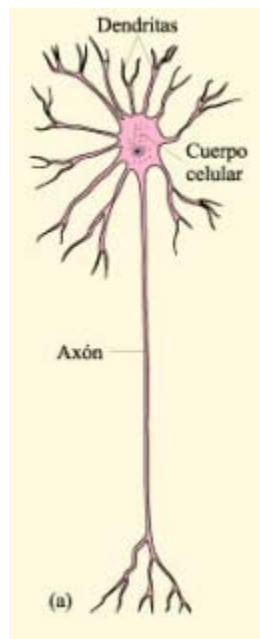


Fig 3.2. Estructura básica de una neurona biológica [NEU]

Se calcula que hay en torno a 10^{12} neuronas en el cerebro de una persona joven, que se comunican entre ellas por medio de procesos electroquímicos.

Cuando una neurona se encuentra en estado de reposo, el exterior de la membrana y el interior se encuentran a diferente potencial eléctrico. La presencia de un estímulo provoca un cambio en esta diferencia, que si alcanza un determinado valor, dispara el conocido como **potencial de acción**: un flujo de iones (corriente eléctrica) que se transmite a través de la neurona desde las dendritas hasta el cuerpo celular, y de ahí a lo largo del axón hasta llegar a su extremo.

Una vez aquí, ya que las neuronas no se encuentran conectadas físicamente entre ellas, si no que existe una pequeña región que las separa, comienza el proceso de la **sinapsis**, que comunica diferentes neuronas entre sí permitiendo que el flujo de información continúe viajando. Se produce por la generación de unas sustancias químicas llamadas neurotransmisores en el extremo de los axones de la primera neurona, que se liberan y

viajan hasta una neurona contigua. Estos neurotransmisores se unen a los receptores: moléculas en la membrana de la siguiente neurona. Al unirse el neurotransmisor con su receptor se producen ciertos cambios, como la apertura de un canal para el paso de iones y otro potencial de acción en la siguiente neurona (entrada excitadora); aunque en otros casos la unión del neurotransmisor puede producir otros efectos, como dificultar que se produzca el potencial de acción en la siguiente neurona (entrada inhibitoria).

3.3 Tipos de redes neuronales artificiales

Las redes más empleadas en la actualidad son las siguientes:

- Perceptrón multicapa: Probablemente el modelo más estudiado y con mayor número de aplicaciones, principalmente para resolución de problemas de modelización, clasificación y predicción. Es el tipo de red en el que nos centraremos en este proyecto.
- SOM: Red basada en base a un modelo biológico, cuya función principal es el agrupamiento de datos automático (son redes que aprenden solas, sin la necesidad de presentarles a priori patrones con la salida deseada).
- Red de Hopfield: Tuvo una gran importancia histórica, ya que supuso el resurgimiento de las redes neuronales tras un parón de casi 20 años. Son redes realimentadas, donde la salida de las neuronas pasa a ser una nueva variable de entrada. Se aplican como memoria asociativa y en problemas de optimización.
- Otras: Existen muchos más modelos con diferentes arquitecturas y algoritmos. Entre ellos cabe destacar las máquinas de vectores soporte (SVN), o las redes de base radial (RBF), derivadas del perceptrón multicapa y con aplicaciones similares.

3.4 Fundamentos generales del proceso de aprendizaje

Un elemento clave en estos sistemas es el proceso de **entrenamiento o aprendizaje**. Para que la red sea capaz de producir la salida deseada, es necesario que aprenda qué se espera de ella, generalmente introduciendo **combinaciones entrada-salida_deseada** (También existen redes que se entrenan “solas” como veíamos antes, sin presentar los patrones de salida deseados, y que se aplican generalmente a problemas de clasificación de patrones).

Pero veamos primeramente esto con un ejemplo en Matlab, aplicándolo al ajuste de una función con un ruido aleatorio.

Comenzamos creando la función, en este caso una parábola con una componente aleatoria de ruido, y la dibujamos.

```
>> x=-2:0.05:2;  
>> y=x.^2+randn(size(x))*0.1;  
>> y_teorico=x.^2;  
>> plot(x,y,'red',x,y_teorico,'black');
```

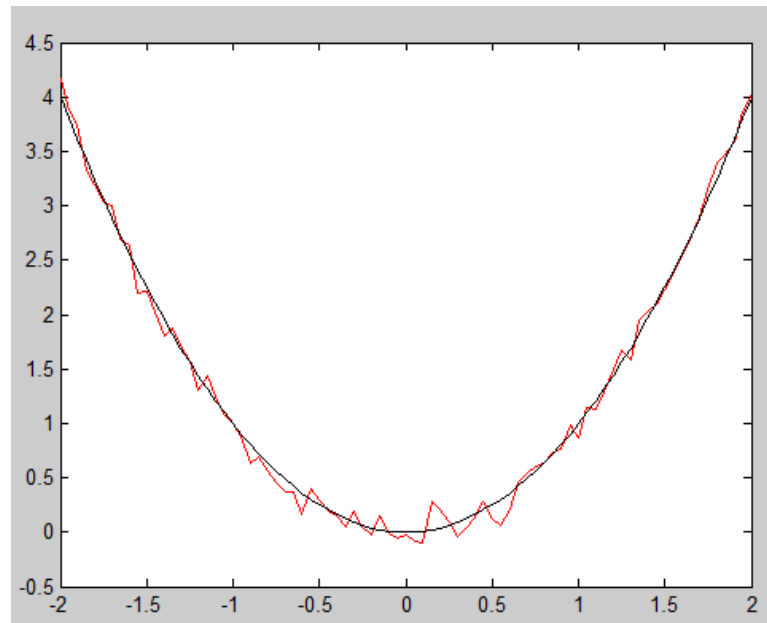


Fig 3.3. Función con ruido aleatorio (rojo) frente a función sin ruido (negro). Elaboración Propia

Ahora vamos a crear una red neuronal, con 1 capa oculta de 5 neuronas y parámetros de entrenamiento por defecto:

```
>> net=newff(x,y,5);
```

Aunque le introduzcamos como parámetro la salida buscada, esto tan sólo es para que la red configure sus dimensiones, no realiza ningún cálculo. Vamos a simular directamente la salida de la red:

```
>> y_net=sim(net,x);  
>> plot(x,y_net,'green',x,y_teorico,'black');
```

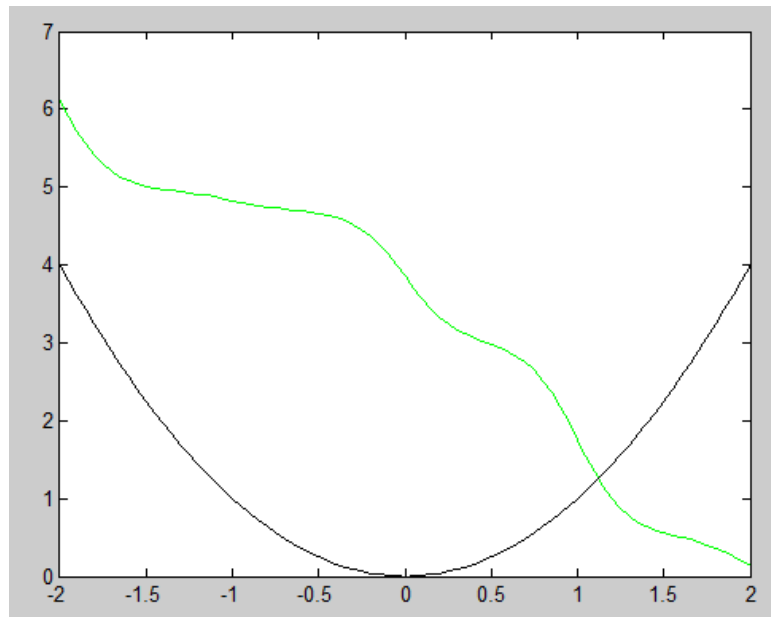


Fig 3.4. Resultado red sin entrenar (verde) y datos sin ruido (negro). Elaboración Propia

Como puede observarse en la gráfica anterior, la salida de la simulación no tiene ninguna relación con la función buscada. Esto es debido a que la red no ha sido entrenada. Al igual que si a un niño le damos números, y le decimos que los multiplique entre ellos sin haberle enseñado a multiplicar, si a una red le damos unos datos, y no le hemos enseñado que hacer con ellos, producirá una salida sin criterio alguno.

Pasemos entonces a entrenar a la red. Para ello llamamos a la función ‘train’ con los parámetros por defecto, que presentará a la red un conjunto de valores de entrada con su correspondiente salida esperada, y simulemos después a ver si hemos mejorado.

```
>> net=train(net,x,y);
>> y_net=sim(net,x);
>> plot(x,y_net,'green',x,y_teorico,'black');
```

Como puede observarse en la siguiente figura, tras el proceso de entrenamiento, la salida de la red tiende a ajustar los valores de la función original, disminuyendo notablemente el componente de ruido.

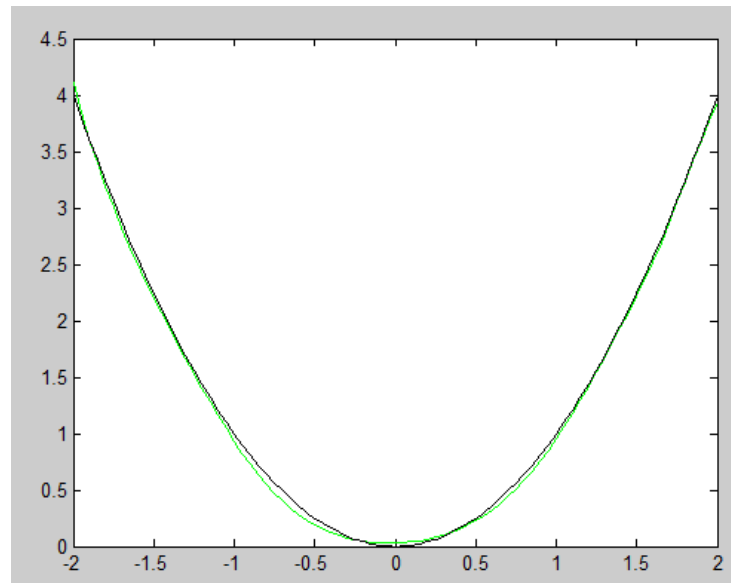


Fig 3.5. Resultado red entrenada (verde) frente a los datos sin ruido (negro). Elaboración Propia

Con este simple ejemplo, podemos empezar a ver el potencial de este método. En el transcurso de los siguientes apartados, trataremos de entender cómo la red es capaz de extraer la información principal de los patrones que se le presentan, eliminando el ruido de los mismos. Como adelanto, entre los valores por defecto de configuración de la red, figura el ‘early stopping’, que combinado con el bajo número de neuronas con el que creamos la red, permite que la red no ajuste uno a uno los patrones de entrenamiento (“overfitting”) y sea capaz de obtener la tendencia de los mismos.

Nota: De ahora en adelante, habremos de tener en cuenta que, debido a la aleatoriedad del proceso de inicialización de los pesos de la red, nunca obtendremos exactamente los mismos resultados si repetimos estos pasos. No obstante, si el lector tiene interés y reproduce el código en su ordenador, verá que no hay variaciones excesivas entre unas simulaciones y otras.

Probemos a entrenar ahora la red con dos pequeñas modificaciones: aumentemos el número de neuronas en la capa oculta hasta 30, y quitemos el stop por validación (ahora veremos lo que significa).

```
>> net=newff(x,y,30);
>> net.divideFcn="";
```

Si volvemos a entrenar la red y simular la salida, exactamente como hicimos antes, vemos lo siguiente:

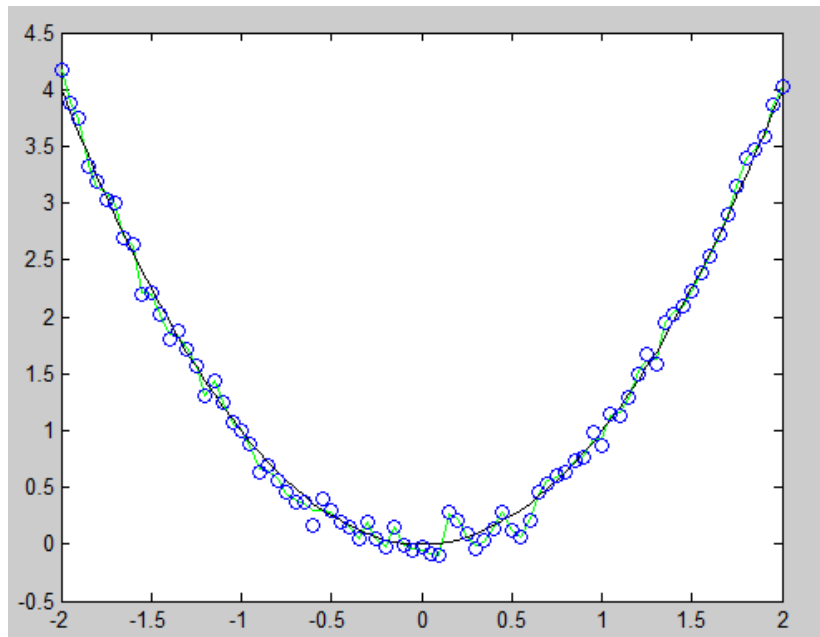


Fig 3.6. Resultado red entrenada (verde) frente a función sin ruido (negro), y los datos con ruido representados con círculos (azules). Elaboración Propia

¡La red ha dejado de quitar el ruido! Lo que hace ahora mismo es reproducirlo casi a la perfección, fijémonos como la curva verde que representa la salida de la red pasa prácticamente por todos los valores de la función con ruido representados con círculos.

Hay dos factores que han provocado esto, justo las modificaciones que realizamos antes de entrenar la red. Al tener un mayor número de neuronas, la red tiene muchos más parámetros disponibles para ajustarse, y como lo que tiende a hacer durante el entrenamiento es a minimizar el error entre su salida y la salida deseada (que son los datos con ruido), si tiene la posibilidad, tenderá a reproducirlos de la manera más precisa que pueda.

Este fenómeno es conocido como “**Sobreentrenamiento**” y para muchas aplicaciones no es algo deseable, lo que queremos en general es precisamente lo contrario: que la red extraiga la tendencia de los datos y sea capaz de responder de una forma lógica cuando se le presenten datos que no ha visto con anterioridad durante el entrenamiento. A esto se le llama capacidad de “**Generalización**”.

Precisamente para mejorar esta capacidad de generalización, Matlab divide por defecto aleatoriamente los datos en tres grupos:

- **Datos de entrenamiento:** Los que se usan para ajustar los pesos y minimizar el error durante el entrenamiento.
- **Datos de validación:** Los que se usan para estimar en cada iteración el error de generalización, de manera que si crece un número determinado de veces (6 por defecto) de manera consecutiva, se detiene el entrenamiento y la red se queda en la configuración con menor error en los datos de validación.
- **Datos de Test:** Utilizados para estimar el error de generalización al final del proceso. La red no los utiliza durante el entrenamiento.

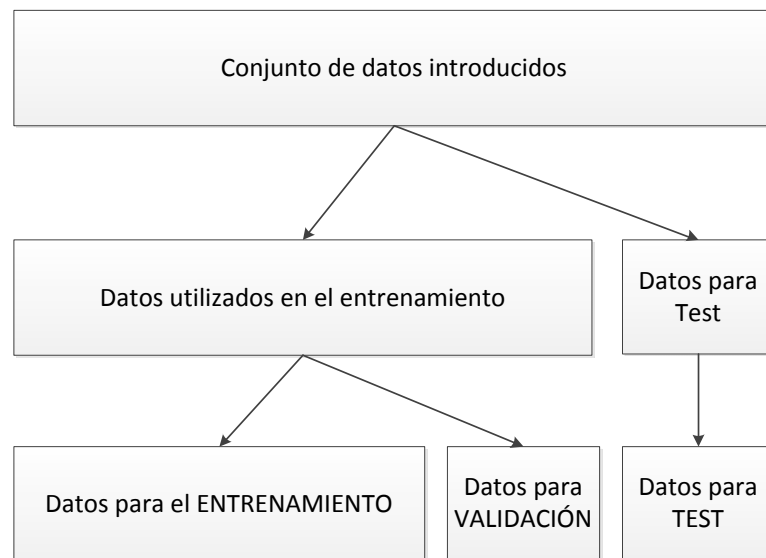


Fig 3.7. Esquema de la división del conjunto de datos para mejorar la capacidad de generalización. Elaboración Propia

Mediante la otra instrucción que modificamos (`net.divideFcn=""`), lo que hicimos precisamente fue desactivar esta división de los datos, con lo que todos los valores son usados para entrenar a la red.

Mostremos ahora la salida del proceso de entrenamiento que nos ofrecía Matlab cuando entrenamos la red, comparando ambos casos:

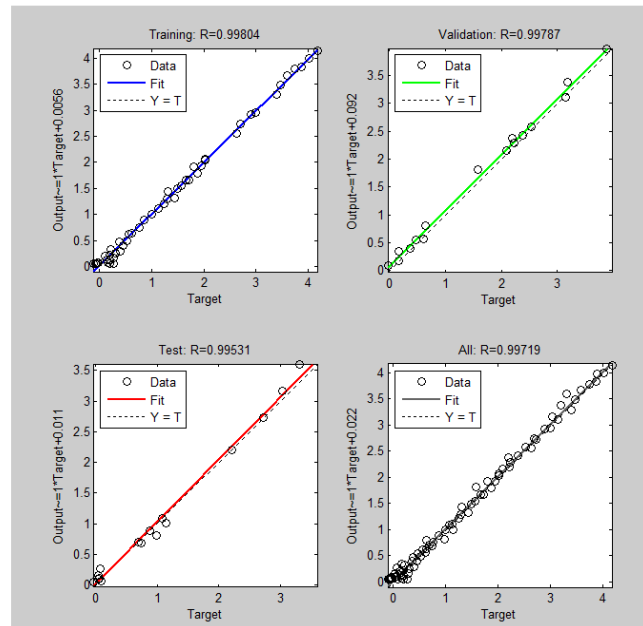
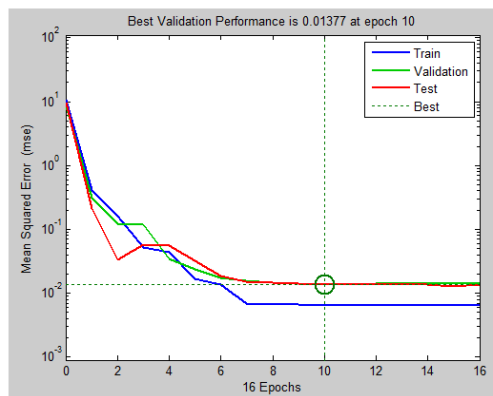


Fig 3.8. Resultados de entrenamiento con 5 neuronas y Validation Stop

Como puede observarse, Matlab divide los datos y evalúa cual es el error que estamos cometiéndolo en cada iteración (gráfico de la izquierda), de manera que cuando el error de validación comienza a crecer en la iteración 10, y lo hace 6 veces consecutivas, detiene el entrenamiento ofreciendo como mejor resultado el que se presentó en la iteración 10.

Veamos qué ocurre en el siguiente caso, con 30 neuronas y sin parada por validación:

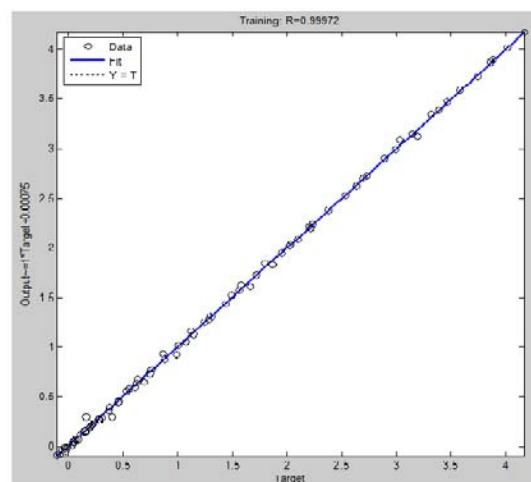
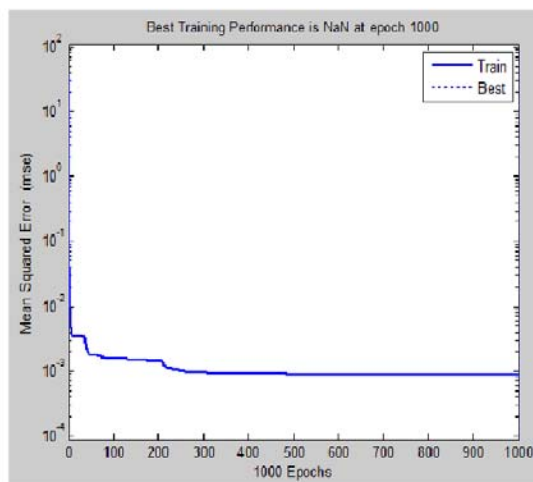


Fig 3.9. Resultados de entrenamiento con 30 neuronas sin Validation Stop

Ahora la red ya no divide los datos y tan sólo contiene un set de entrenamiento, que usará hasta alcanzar el número máximo de iteraciones permitido (1000 en este caso).

Los gráficos de la derecha muestran las regresiones entre los valores introducidos como salida deseada y la salida que simula la red una vez entrenada. Si el ajuste fuese perfecto, tendría como ecuación $y=x$, y $R=1$. Vemos como los datos de entrenamiento (Training) están bastante más correlacionados en el segundo caso... ¡Claro! La red tiende a ajustar los datos con ruido casi de manera perfecta, ¡que era precisamente lo que no buscábamos!

3.5 Componentes de la neurona artificial

El sistema más simple es el que tan sólo contiene una neurona, como se muestra a continuación:

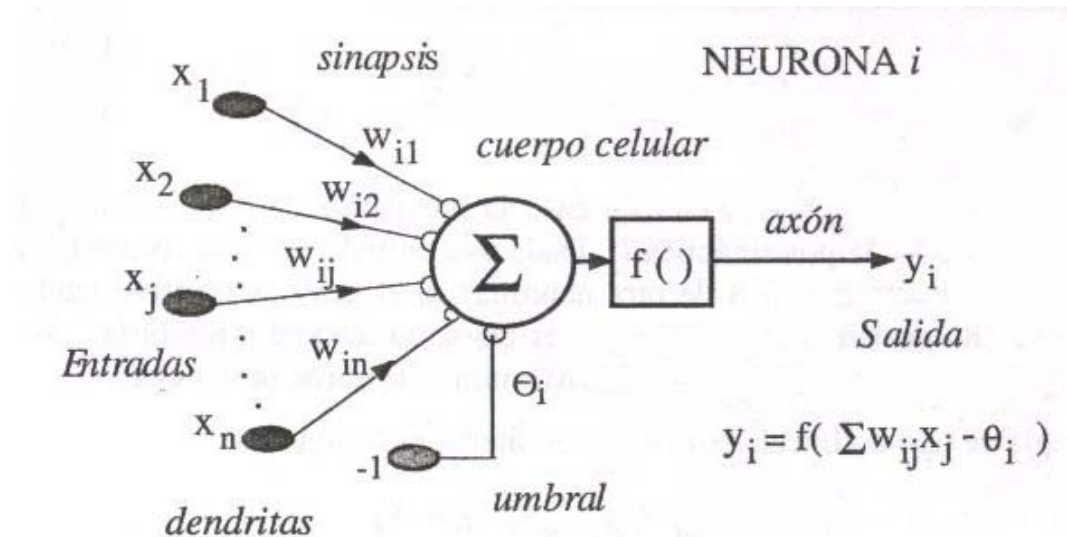


Fig 3.10 Modelo de neurona estándar [MS]

Pasemos a describir sus componentes y funcionamiento.

Los valores x_i , con $i=1, \dots, n$ son las **entradas** a la red,

Con frecuencia, se añade un grado de libertad adicional θ , conocido como el **umbral**, que funcionan a modo de puerta para neuronas todo-nada, y representará en este caso el valor de activación de la neurona. En muchas ocasiones se considerará, tal y como haremos nosotros, como una nueva entrada a la red x_0 de valor constante.

Los w_{ij} son llamados **pesos sinápticos**, que multiplican a las entradas emulando el papel de la sinapsis biológica.

Una **regla de propagación**:

$$h_i(t) = \sigma[w_{ij}, x_j(t)]; \text{ habitualmente } h_i(t) = \sum w_{ij} \cdot x_j(t) \quad (\text{ec 3.1})$$

De ahí, este valor pasa a través de una **función de activación** que llamaremos ϕ_m , generalmente no lineal, para producir la salida de la neurona:

$$y_i(t) = f_i(h_i(t)) = f\left(\sum_{j=1}^n w_{ij} \cdot x_j(t) - \theta\right) = f\left(\sum_{j=0}^n w_{ij} \cdot x_j(t)\right) \quad (\text{ec 3.2})$$

Se muestran a continuación algunas funciones de activación más usuales:

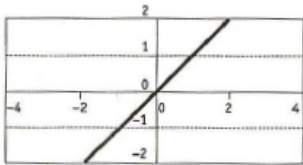
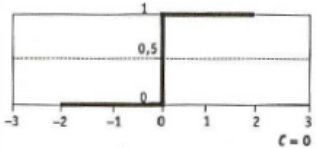
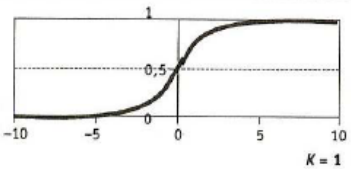
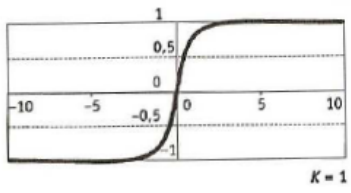
<p>Función de transferencia identidad o lineal:</p> $a_j = f[Net_j(t)] = Net_j$ <p>Rango = $(+\infty, -\infty)$</p>	
<p>Función de transferencia escalón:</p> $a_j = f(Net_j) = \begin{cases} 1 & \text{si } Net_j \geq c \\ 0 & \text{si } Net_j < c \end{cases}$ <p>Rango = $\{0,1\}$</p>	
<p>Función de transferencia sigmoide o logística:</p> $a_j = f(Net_j) = \frac{1}{1 + e^{-k \cdot Net_j}}$ <p>Rango = $(0,1)$ $k > 0$</p>	
<p>Función de activación tangente, tangente hiperbólica o tangencial:</p> $a_j = f(Net_j) = \frac{e^{k \cdot Net_j} - e^{-k \cdot Net_j}}{e^{k \cdot Net_j} + e^{-k \cdot Net_j}}$ <p>Rango = $(-1,1)$ $k > 0$</p>	

Fig 3.11. Funciones de activación [Izq]

3.6 ADALINE (ADaptive LInear NEuron)

Una Adalina, es una neurona elemental con una función de activación de tipo lineal $f(x)=x$, con lo que nos queda la salida de la neurona:

$$y_i(t) = f_i(h_i(t)) = f\left(\sum_{j=1}^n w_{ij} \cdot x_j(t) - \theta\right) = \sum_{j=1}^n w_{ij} \cdot x_j(t) - \theta \quad (\text{ec 3.3})$$

Otra característica que define a la Adalina, es la regla de aprendizaje, en este caso la regla de Widrow-Hoff, o regla LMS (Least Mean Squares), que actualiza los pesos sinápticos de manera proporcional al error que se comete durante el proceso de aprendizaje, calculado como diferencia entre la salida simulada y la salida esperada del patrón de entrenamiento.

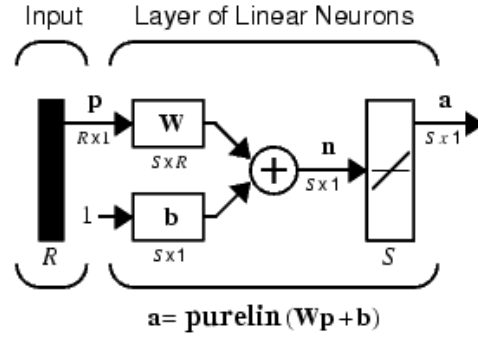


Fig 3.12. Estructura de la Adalina [MAT]

Según vemos en la figura superior, consideraremos una capa de entrada con R neuronas, y una capa de salida con S.

El algoritmo de aprendizaje LMS

Como comentábamos al comienzo del capítulo, para que una red funcione correctamente tiene que seguir un proceso de aprendizaje, en el que se le presentan unos patrones entrada-salida_deseada. Vamos a ver ahora cómo funciona el algoritmo interno de aprendizaje que utiliza la Adalina, el LMS, base del algoritmo BackPropagation que veremos más adelante.

Empecemos diciendo que este algoritmo es uno de los denominados “por corrección de errores”. Los algoritmos de este tipo ajustan el valor de los pesos proporcionalmente a la diferencia (error) entre la salida simulada por la red y la salida deseada o “Target”.

El método consiste en definir una función de error (E), que dependerá del valor de los pesos sinápticos (W), y que optimizaremos (en general, minimizaremos) habitualmente por el método conocido como descenso por gradiente. Este método nos proporcionará una regla para actualizar los pesos en cada iteración, hasta que lleguemos al mínimo absoluto de la función de error, que provendrá del conjunto de pesos óptimos para la red.

Sea $\vec{x}^k, k = 1 \dots p$ el conjunto finito de “p” patrones de entrada, con sus respectivas salidas

$$E: \mathbb{R}^{R \times S} \rightarrow \mathbb{R}$$

\vec{t}^k . Entonces, definamos una función

$$W \rightarrow E(W) ,$$

y una regla de actualización de los pesos

$$W(t+1) = W(t) - \varepsilon \cdot \nabla E(W(t)) \quad (\text{ec } 3.4)$$

Es fácil comprobar calculando la diferencial del error y aplicando la regla de la cadena, que con este método siempre nos dirigimos hacia un mínimo, al movernos en el sentido contrario al gradiente. Lo que no queda garantizado en el caso general es que nos conduzca a un mínimo global (sí que lo es en el caso de la Adalina, ya que al tener una función de activación lineal, el error es un polinomio de grado 2 en los pesos, un paraboloide, que tiene uno y sólo un mínimo local, y que también es global, por lo que el método expuesto nos conduce directamente hacia él).

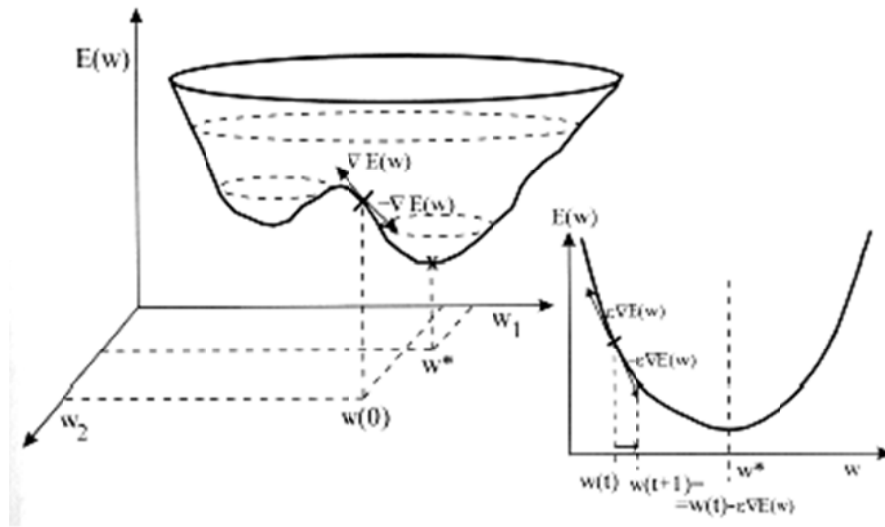


Fig 3.13. Ilustración del método de descenso por gradiente [MS]

En este punto, sólo nos queda por ver cuál será la función de error, y realizar los cálculos que nos conduzcan al criterio de actualización de los pesos sinápticos. Elijamos una de las más comunes, cuadrática en el error:

$$E(w_{ij}) = \frac{1}{2} \sum_{k=1}^p \sum_{i=1}^S (t_i^k - y_i^k)^2 \quad (ec \ 3.5)$$

Cabe mencionar que existen otras muchas funciones de error a aplicar dependiendo del problema. Por ejemplo, la función entrópica, derivada de la física estadística, es muy usada cuando los patrones son binarios. En nuestro caso, ya que la función de error refleja de alguna manera la distribución del ruido presente en los datos, suponemos que la mejor aproximación será asumir que cada variable se distribuye de manera independiente y Gaussiana, aceptando entonces como función de error la propuesta, una suma de cuadrados.

Si derivamos la función de error con respecto a los pesos, teniendo en cuenta que la salida de la red para cada patrón de entrenamiento es:

$$y_i^p = \sum_{j=1}^p w_{ij} x_j^p - \theta \quad (ec \ 3.6),$$

y que las derivadas parciales hacen que los valores diferentes de (i, j) desaparezcan en las sumas, obtenemos que:

$$\frac{\partial E}{\partial w_{ij}} = -\left(\frac{1}{2}\right) \cdot 2 \cdot \sum_{k=1}^p (t_i^k - y_i^k) \frac{dy}{dw_{ij}} = -\sum_{k=1}^p (t_i^k - y_i^k) x_j^p \quad (ec \ 3.7)$$

Si volvemos a la expresión originaria del método ec.3.4, y sustituimos esto en el gradiente, tenemos que cada peso se actualizará de la siguiente manera:

$$\Delta w_{ij} = -\varepsilon \frac{\partial E}{\partial w_{ij}} = \varepsilon \sum_{k=1}^p (t_i^k - y_i^k) x_j^k \quad (\text{ec 3.8})$$

Si analizamos la expresión anterior, vemos que la variación de cada peso es proporcional a la suma de los errores cometidos al presentar los patrones a la red, ponderados por los correspondientes valores de la entrada.

La constante de proporcionalidad ε es conocida como ritmo de aprendizaje, y marca cómo de grande es la actualización de los pesos. Recordemos que aparecía por primera vez multiplicando al gradiente en la expresión ec3.4, ahí es donde vemos que lo que hace es marcar la longitud de cada tramo del camino que seguimos, desde el punto inicial en el espacio de los pesos, hasta que alcancemos el óptimo. Por tanto, su valor ideal sería infinitesimal, es decir, un valor pequeño implica un aprendizaje lento pero seguro, mientras que uno grande puede llevar a oscilaciones durante el entrenamiento por dar vueltas en torno al óptimo.

3.7 El Perceptrón Multicapa (MLP)

En los dos anteriores apartados, hemos pasado del detalle de una neurona individual, a la Adalina, una red básica que conecta las neuronas de entrada con las de salida y posee en las últimas una función de activación lineal. Vamos ahora a generalizar lo anterior:

Entre la entrada y la salida, tendremos una o más capas de neuronas, llamadas capas ocultas, con sus correspondientes pesos sinápticos y funciones de activación.

Estas funciones de activación ya no serán necesariamente lineales.

Extenderemos la regla LMS para este caso, obteniendo el algoritmo de Back-Propagation.

Con estas definiciones, tendremos una red que se asemejará a la de la siguiente figura:

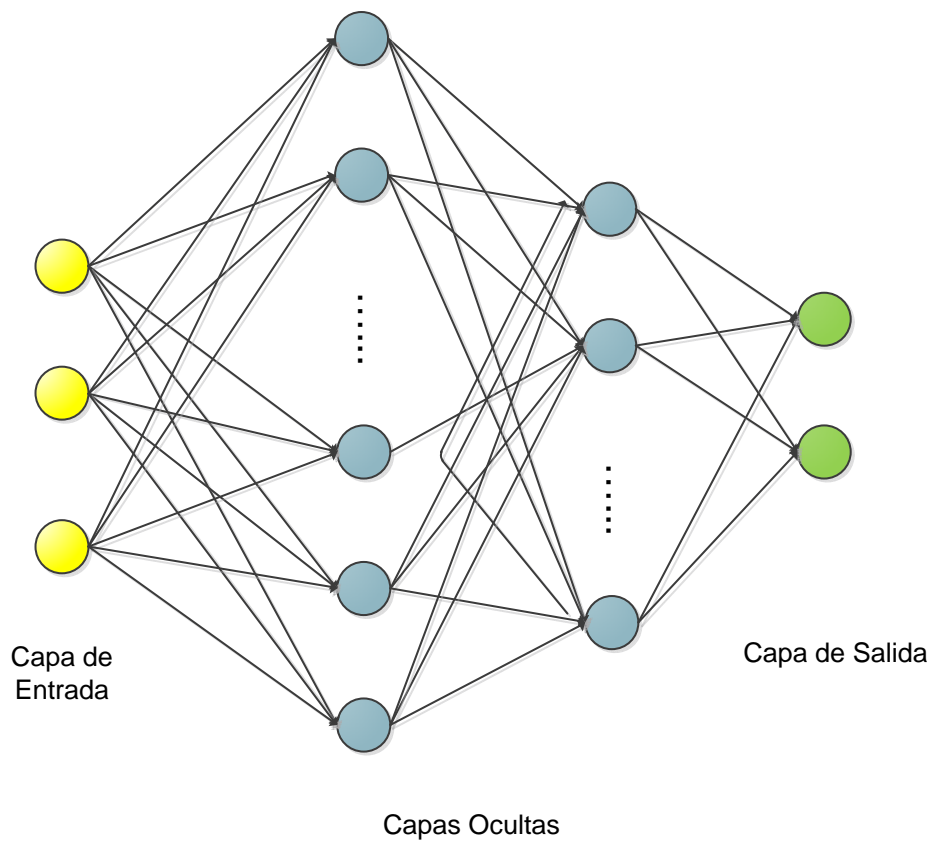


Fig 3.14. Estructura general de una red de Perceptrón Multicapa. Elaboración Propia

En este caso, tenemos una red con 2 capas ocultas, un número arbitrario de neuronas en cada una de ellas, 3 neuronas en la capa de entrada, y 2 en la de salida. Consideremos de ahora en adelante, por simplicidad, una red como la de la figura anterior, pero con sólo 1 capa oculta.

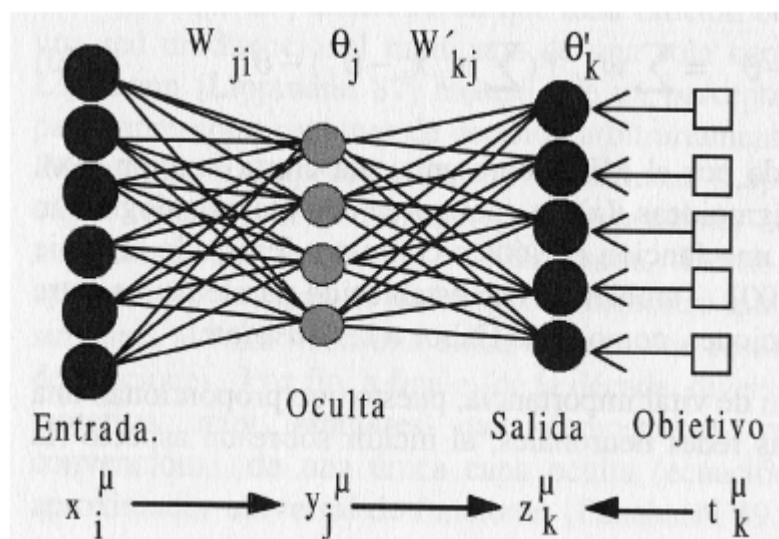


Fig 3.15. Esquema de un perceptrón multicapa de una capa oculta [MS]

Denominaremos en este caso x_i a las entradas a la red, y_j a las salidas de la capa oculta, y z_k a las de la capa final (y por tanto, salidas de la red). Por otro lado, t_k serán las salidas deseadas para el proceso de entrenamiento, w_{ji} los pesos de la capa oculta (que conectan las neuronas de entrada con la capa oculta), θ_j sus umbrales, w'_{kj} los pesos de la capa de salida (que conectan la capa oculta con la salida), y θ'_k sus umbrales.

Pasemos entonces a ver qué salida producen las neuronas de la capa de salida al introducir un dato de entrada:

$$z_k = g \left(\sum_j w'_{kj} y_j - \theta'_k \right) = g \left(\sum_j w'_{kj} \cdot f \left(\sum_i w_{ji} x_i - \theta_j \right) - \theta'_k \right) \quad (ec \ 3.9)$$

Donde $g(s)$ es la función de activación en la capa de salida, que en general suele tomarse una función lineal $g(s)=s$ que deja invariante a su argumento y por lo tanto no aparecería en la expresión anterior, y como función de activación de la capa oculta una genérica $f(x)$. Como único requisito para estas funciones, el que han de ser una de clase $C^1(\mathbb{R})$ si queremos utilizar el algoritmo de aprendizaje Back-propagation que se expone a continuación, ya que involucra derivadas de las mismas. Pasemos entonces a entrenar a la red.

El algoritmo de aprendizaje Back-Propagation

Sea $\vec{x}^\mu, \mu=1\dots p$ el conjunto finito de “p” patrones de entrada, con sus respectivas salidas \vec{t}^μ .

Definimos la función de error de manera análoga a como lo hacíamos para el algoritmo LMS a modo de suma de cuadrados, sólo que ahora aparecen muchos más parámetros a tener en consideración.

$$E(W) = \frac{1}{2} \sum_\mu \sum_k \left(t_k^\mu - g \left(\sum_j w'_{kj} y_j^\mu - \theta'_k \right) \right)^2 \quad (ec \ 3.10)$$

Y, al igual también que en el LMS, emplearemos el descenso por gradiente, de manera que la regla de actualización de los pesos tendrá la forma de:

$$\Delta w'_{kj} = -\varepsilon \cdot \frac{\partial E}{\partial w'_{kj}} \quad (ec \ 3.11)$$

$$\Delta w_{ji} = -\varepsilon \cdot \frac{\partial E}{\partial w_{ji}} \quad (ec \ 3.12)$$

Pasemos a calcular ahora el gradiente y obtener la expresión buscada, derivando tal como lo hicimos anteriormente, aplicando sucesivamente la regla de cadena:

$$\Delta w'_{kj} = \varepsilon \sum_{\mu} \delta'^{\mu}_k \cdot y^{\mu}_j; \text{ con } \delta'^{\mu}_k = [t^{\mu}_k - g(h^{\mu}_k)] \frac{\partial g(h^{\mu}_k)}{\partial h^{\mu}_k}; \text{ siendo } h^{\mu}_k = \sum_j w'_{kj} y^{\mu}_j - \theta'_k \quad (\text{ec 3.13})$$

$$\Delta w_{ji} = \varepsilon \sum_{\mu} \delta^{\mu}_j \cdot x^{\mu}_i; \text{ con } \delta^{\mu}_j = \left(\sum_k \delta'^{\mu}_k w'_{kj} \right) \frac{\partial f(h^{\mu}_j)}{\partial h^{\mu}_j}; \text{ siendo } h^{\mu}_j = \sum_i w_{ji} x^{\mu}_i - \theta_j \quad (\text{ec 3.14})$$

Si nos fijamos en la manera de actualización de los pesos, en la segunda de las ecuaciones anteriores aparece un parámetro que ha de ser calculado con anterioridad en la primera, δ'^{μ}_k , llamada señal de error por ser proporcional directamente al error cometido al presentar los patrones de entrenamiento. Entonces, vemos el porqué del nombre del algoritmo: primero calculamos la actualización de los pesos de la capa de salida, y de ahí este valor de la señal de error se propaga hacia atrás para actualizar los pesos de la capa anterior.

3.8 Aplicaciones de las Redes neuronales

Pese a su relativa juventud, estos modelos se aplican ya con éxito en un amplio abanico de áreas. Citemos entre ellas:

- Medicina: Diagnóstico de cardiopatías mediante la clasificación de señales electrocardiográficas, detección de tumores en imágenes médicas, estudio del funcionamiento del cerebro,...
- Procesado de la señal: Construcción de ecualizadores lineales y canceladores de ecos (¡millones de módems llevan una adalina!), eliminación activa del ruido en señales, reconocimiento de patrones,...
- Economía: Decisiones sobre concesión de créditos, determinación de la probabilidad de quiebra de un banco, estimación de la demanda y stocks,...
- Medio ambiente: Predicción de la irradiación solar, de niveles tóxicos de ozono, de variaciones globales de temperatura,...
- Otras: Sistemas de conducción y guiado automáticos, modelos de fenómenos en física estadística,

4 Selección de una red neuronal para simular diámetro y velocidad en un lecho experimental

4.1 Obtención de los datos experimentales

El punto de partida son 20 conjuntos de datos previamente obtenidos en un lecho fluidizado bidimensional en el laboratorio de Ingeniería Térmica y de Fluidos de la Universidad Carlos III de Madrid.

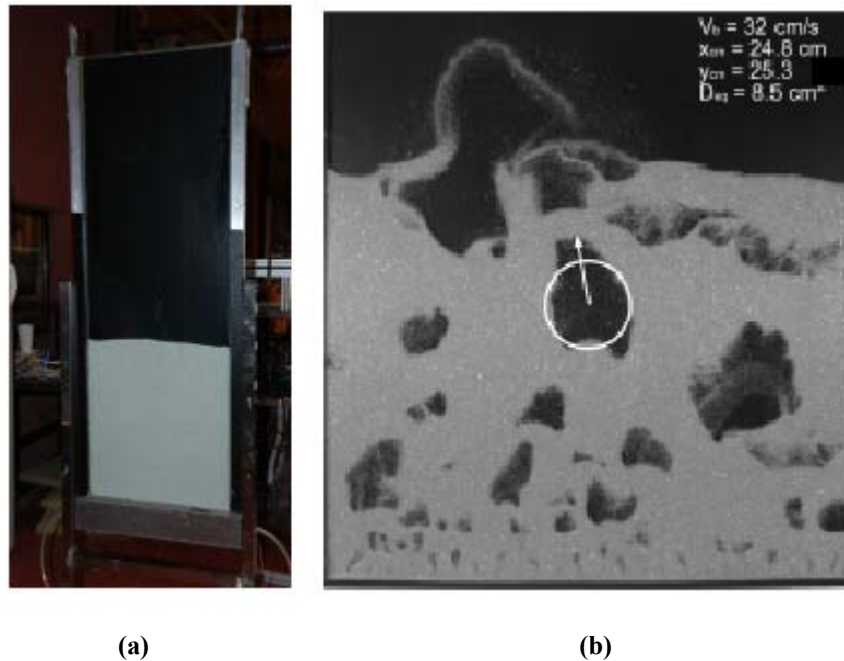


Fig 4.1. (a) Lecho experimental de la UC3M [ISE] (b) Fotografía de burbujas en el lecho [Izq]

A partir de un programa de procesamiento de imágenes, se identifica la posición del centro de masas de las burbujas, así como su diámetro y velocidad estimados. Estos datos son el punto de partida del estudio del presente proyecto, y consisten en 20 set de datos, tomados para distintas alturas de lecho fijo y caudales de entrada, cada uno de ellos conteniendo la siguiente información:

Variables comunes a todos los set de datos

- Posición del centro de masas de la burbuja (X_{cm} , Y_{cm}) relativa a un origen de coordenadas situado en el extremo inferior izquierdo del lecho.
- Diámetro de la burbuja.
- Velocidad de la burbuja (V_x , V_y).





















Variables que diferencian cada set de datos

- Caudal de entrada de aire, equivalente a la relación entre la velocidad del lecho y la velocidad de mínima fluidización.
- La altura de lecho fijo (antes de expandirse y comenzar la fluidización).

Tabla 4.1. Combinaciones de datos utilizados en función de la velocidad y altura de lecho fijo

U/U_{mf}	1.5	1.75	2	2.25	2.5
Altura lecho fijo (cm)	30	40	50	60	

Tabla 4.2. Datos de partida, en la forma “BB_Velo2_altura lecho fijo(cm)_caudal (l/min)”

	bb_Velo2_60_Q_204.mat
	bb_Velo2_60_Q_184.mat
	bb_Velo2_60_Q_164.mat
	bb_Velo2_60_Q_144.mat
	bb_Velo2_60_Q_124.mat
	bb_Velo2_50_Q_184.mat
	bb_Velo2_50_Q_166.mat
	bb_Velo2_50_Q_148.mat
	bb_Velo2_50_Q_130.mat
	bb_Velo2_50_Q_110.mat
	bb_Velo2_40_Q_170.mat
	bb_Velo2_40_Q_154.mat
	bb_Velo2_40_Q_136.mat
	bb_Velo2_40_Q_120.mat
	bb_Velo2_40_Q_102.mat
	bb_Velo2_30_Q_162.mat
	bb_Velo2_30_Q_146.mat
	bb_Velo2_30_Q_130.mat
	bb_Velo2_30_Q_114.mat
	bb_Velo2_30_Q_098.mat

Si combinamos cada valor de U/U_{mf} con los de altura de lecho fijo, obtenemos $5 \times 4 = 20$ conjuntos de datos, en cada uno de los cuáles tenemos en torno a 75.000 registros de burbujas con su posición, diámetro y velocidad.

En la siguiente figura se pueden observar las dimensiones transversales del lecho utilizado en los experimentos:



Fig 4.2. Sección transversal del lecho experimental de la UC3M [Izq]

Cuenta con un distribuidor de 1mm por cada cm de longitud del lecho.

El material sólido son esferas de cristal homogéneas, con buenas propiedades de fluidización, encuadradas dentro de la clasificación GeldartB.

4.2 Análisis y preproceso de los datos experimentales

Me gustaría comenzar este capítulo con una frase del profesor Geldart, extraída de [MSF]: “El tiempo de llegada de una sonda espacial que viaje a Saturno puede determinarse con mucha mayor precisión que el comportamiento de sólidos y fluidos en un lecho fluidizado”. Creo que esta frase nos ayudará de alguna manera a entender la estructura de los datos que se presentan a continuación, y el por qué seleccionamos una red neuronal como modelo matemático de aproximación al problema. Tal y como menciona en el libro ‘Handbook of fluidization and fluid particle systems’ [YAN], algunos autores consideran a los lechos fluidizados como sistemas caótico-deterministas.

Para tener un diseño válido de una red neuronal, es muy importante estudiar qué datos se le introducen a priori, es decir, se hace necesario un análisis previo de las características y medidas estadísticas básicas de los mismos antes de ningún tipo de procesado. Primeramente, dibujaremos el gráfico de dispersión del diámetro de burbuja en función de su posición con respecto al origen:

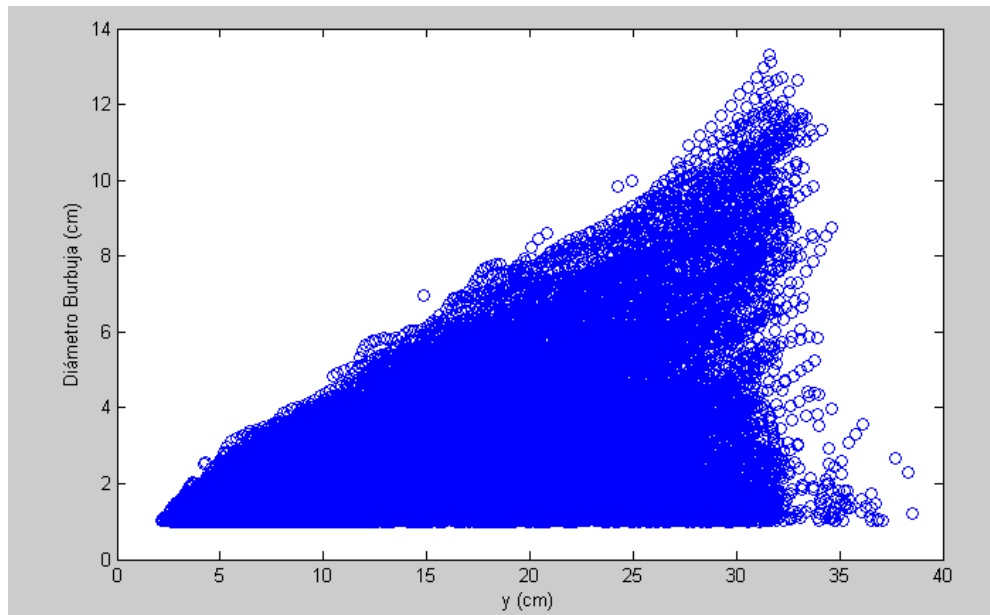


Fig 4.3. Diagrama que muestra la dispersión de los datos. Elaboración Propia

Entonces, teniendo en cuenta que el objetivo final del presente proyecto reside en estimar y predecir el **diámetro medio y velocidad media** de las burbujas, tendremos que considerar una **zona HOMOGÉNEA** del lecho, y por tanto nos vemos obligados a realizar cierto **filtrado de los datos**.

Para ello, observamos los gráficos de dispersión del diámetro de burbuja en función de su posición (tanto Xcm como Ycm) incluidos abajo, que combinado con el conocimiento teórico con el que contamos a priori sobre las características de los lechos fluidizados, nos lleva a que **eliminemos** todas aquellas burbujas que se encuentren en una posición cercana a la región de entrada, evitando la influencia del distribuidor y que las burbujas no estén completamente formadas, y las próximas a altura de lecho fijo (esta es una aproximación conservadora, basada en la incapacidad de estimar a priori la altura del lecho expandido, superior en todo caso a la altura de lecho fijo). Igualmente quitamos todas las burbujas que se encuentren lateralmente cerca de las paredes para evitar su efecto. Se puede observar en la figura 4.5 como cerca de las paredes las burbujas tienen un rango de diámetros diferente al resto (si el centro de masas de una burbuja está a 1 cm de la pared, no puede tener un diámetro superior a 2 cm: estaría fuera del lecho).

Traducido a ecuaciones, nos queda:

- **xinf=5 cm;**
- **xsup=45 cm;**
- **yinf=5 cm;**
- **ysup=h-3 cm; (h = altura de lecho fijo)**

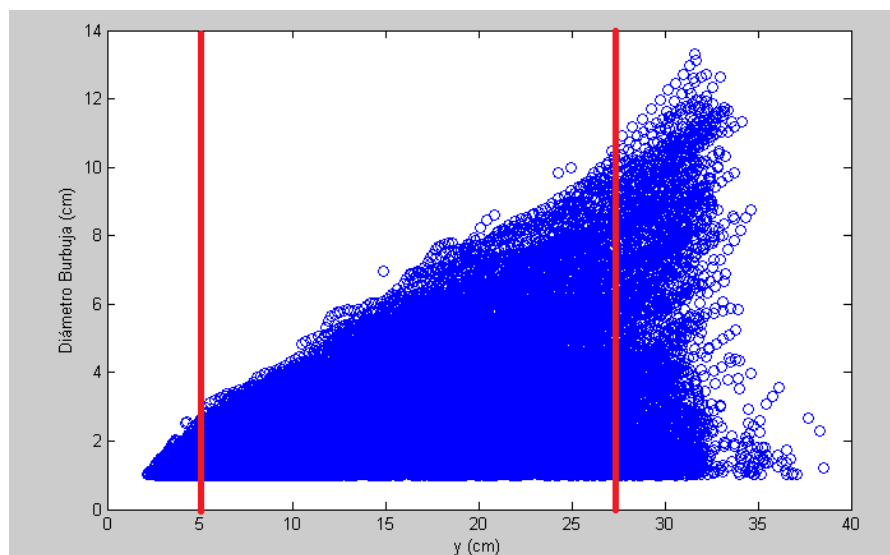


Fig 4.4. Diámetro de la burbuja en función de y (h=30 cm, q=94 l/min)

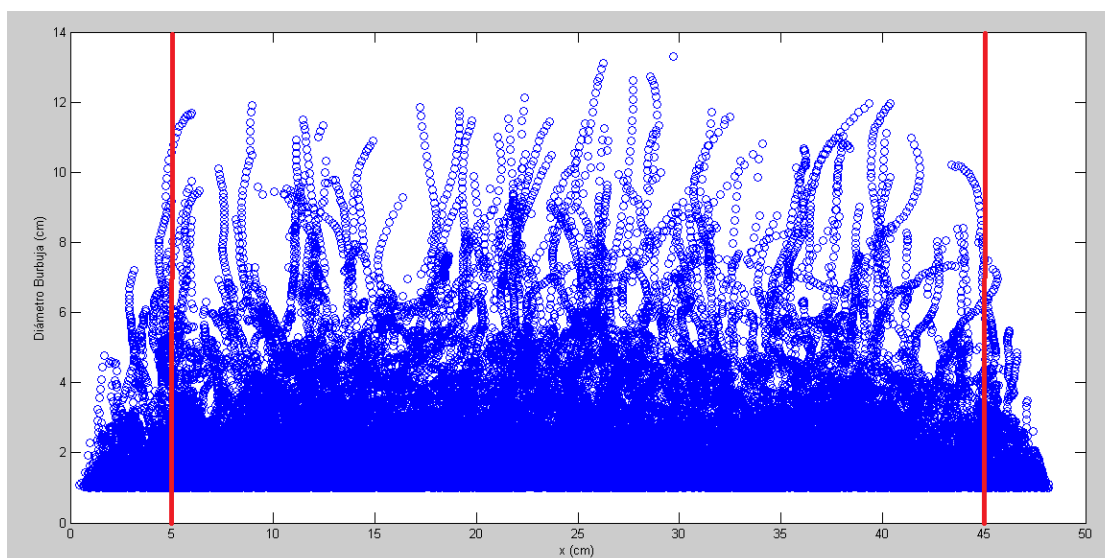


Fig 4.5. Diámetro de la burbuja en función de x (h=30 cm, q=94 l/min)

También eliminamos primeramente ciertos datos atípicos (“outliers”), considerando válidas sólo aquellas burbujas con:

- $D_{min}=1$ cm;
- $D_{max}=15$ cm;
- $V_{ymin}=1$ cm/s;
- $V_{ymax}=100$ cm/s;

El diámetro mínimo de burbuja viene determinado por el espesor del lecho, 0.5cm, por lo tanto si una burbuja tiene un diámetro inferior a 1cm (0.5cm de radio) habría sido mal procesada por el programa de reconocimiento y tratamiento de imágenes.

En cuanto al límite superior para el diámetro, utilizamos como primera aproximación la ec.2.10 adecuándola los datos de que disponemos:

$$d_{b,\max} = 0.65 \left[A(u_0 - u_{mf}) \right]^{0.4} = 0.65 \left[A \left(\frac{Q}{A} - \frac{Q}{u_{mf} A} \right) \right]^{0.4} = 0.65 \left[Q \left(1 - \frac{1}{u_{mf}} \right) \right]^{0.4} \quad (ec \ 4.1)$$

Sustituyendo los valores que tenemos en nuestro caso, por ejemplo los obtenidos con el caudal más bajo y el más alto:

- $q=98 \text{ l/min}$ con $U/U_{mf}=1.5$: $d_{\max}=8.08 \text{ cm}$
- $q=204 \text{ l/min}$ con $U/U_{mf}=2.5$: $d_{\max}=13.7 \text{ cm}$

Si vamos entonces a los gráficos de dispersión, vemos como hay muchas burbujas que quedarían fuera utilizando estos criterios, ya que las correlaciones están basadas en lechos tridimensionales, tal como se comentaba en el capítulo 2. Por lo tanto, como primera aproximación para eliminar ciertos datos atípicos, tomamos un valor común para todos los casos de 15cm, basado en un análisis gráfico y tomando como referencia orientativa el mayor valor del diámetro máximo calculado arriba.

La razón para quitar los datos con V_y negativa o cercana a cero, la encontramos en los procesos de coalescencia y rotura de burbujas. El flujo de aire es hacia arriba, tal y como debiera ser el movimiento de las burbujas, que nacen en el distribuidor y van progresivamente aumentando de tamaño debido a la disminución de presión. Si encontramos burbujas con componente vertical de la velocidad negativa es debido al código de procesamiento de imágenes utilizado, ya que tiene en cuenta para el cálculo de la velocidad la variación del centro de masas de las partículas. Cuando dos burbujas se juntan, si una de ellas tiene el centro de masas situado en una coordenada vertical superior al de la otra, el centro de masas del sistema conjunto disminuye su coordenada vertical en el siguiente instante de tiempo, con lo cual nos queda una sola burbuja con velocidad negativa.

Con los valores excesivos de la velocidad asumimos, utilizamos la ec 2.8. también de manera orientativa, y sustituyendo con el valor del diámetro máximo permitido antes expuesto, 15cm, obtenemos que para una burbuja aislada:

$$u_{br} = 0.711 \sqrt{g d_b} = 0.711 \sqrt{980(\text{cm/s}) \cdot 15\text{cm}} = 86.2 \text{ cm/s} \quad (ec \ 4.2)$$

Con este resultado, y a la vista de las gráficas experimentales, decidimos utilizar un valor de 100 cm/s como límite de velocidad vertical de las burbujas.

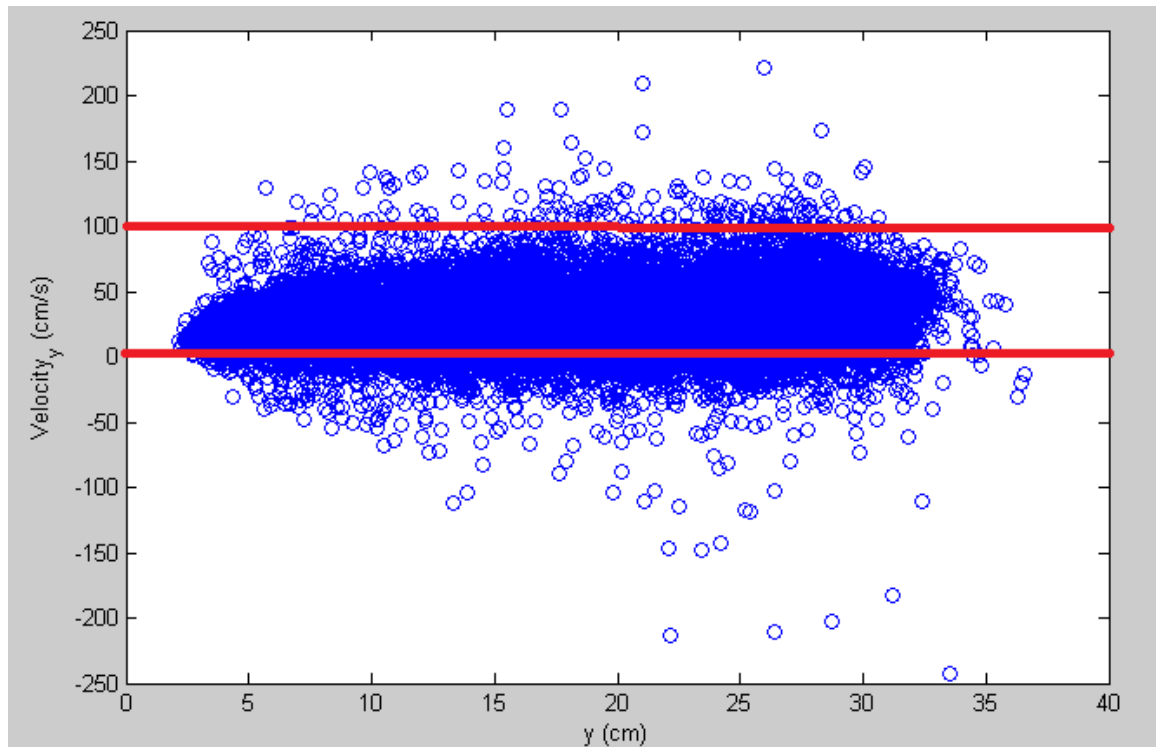


Fig 4.6. Componente vertical de velocidad de la burbuja en función de y (h=30 cm, q=94 l/min)

Resaltemos en este momento que el proceso de filtrado previo que acabamos de comentar marca un rango de valores permitidos de las variables para las cuáles este trabajo es aplicable.

Sería recomendable en el futuro realizar una poda de los datos por velocidad máxima y diámetro máximo que vaya más allá del análisis gráfico realizado, teniendo en cuenta cómo se generaron estos datos por reconocimiento de imágenes y la particularidad de cada set de datos y no un valor común de filtrado para todos ellos, con el objetivo de identificar con mayor claridad los datos que quremos que queden fuera de estudio y ampliar su rango de aplicación.

Una vez hecho esto, representemos ahora la tendencia y dispersión de los datos mediante un BoxPlot, herramienta estadística que representa dentro de la caja al 50% de los datos que quedan en el centro de la distribución, y mediante 2 barras laterales que salen de cada extremo, el 50% restante, esto es, los percentiles 25 y 75. En cruces rojas los datos que el modelo encuentra atípicos y con una línea roja también dentro de la caja el valor de la mediana.

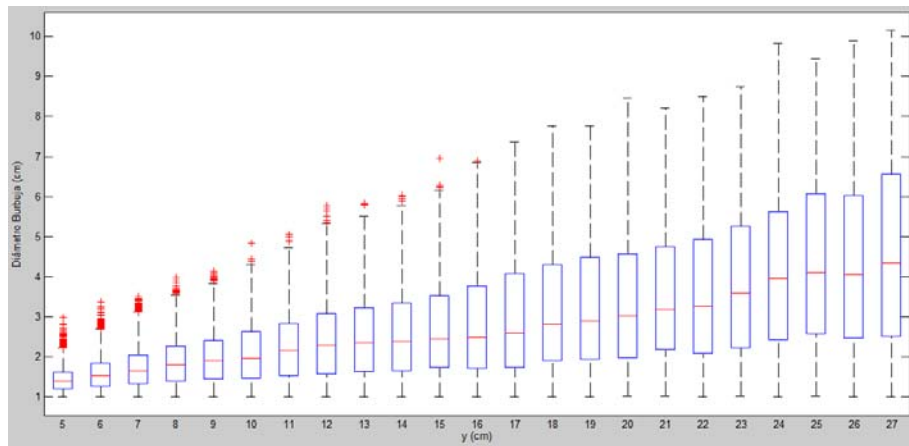


Fig 4.7. BoxPlot para el diámetro de burbuja en función de la altura

Como puede observarse, existe una tendencia creciente en el diámetro “mediano” de la burbuja, así como en la varianza de los datos (el tamaño de las cajas y las barras aumenta conforme subimos por el lecho).

Se hace necesario por tanto otro **filtrado** de los datos, tomando tan sólo como válidas para entrenar la red aquellas burbujas que **en cada altura** (discretizada) caen **dentro de un intervalo de confianza para el diámetro medio de burbuja**, basado en la hipótesis de población desconocida y muestra grande ($n > 100$), que como vemos en el siguiente histograma, con la altura discretizada con amplitud $dy=1$, siempre se cumple:

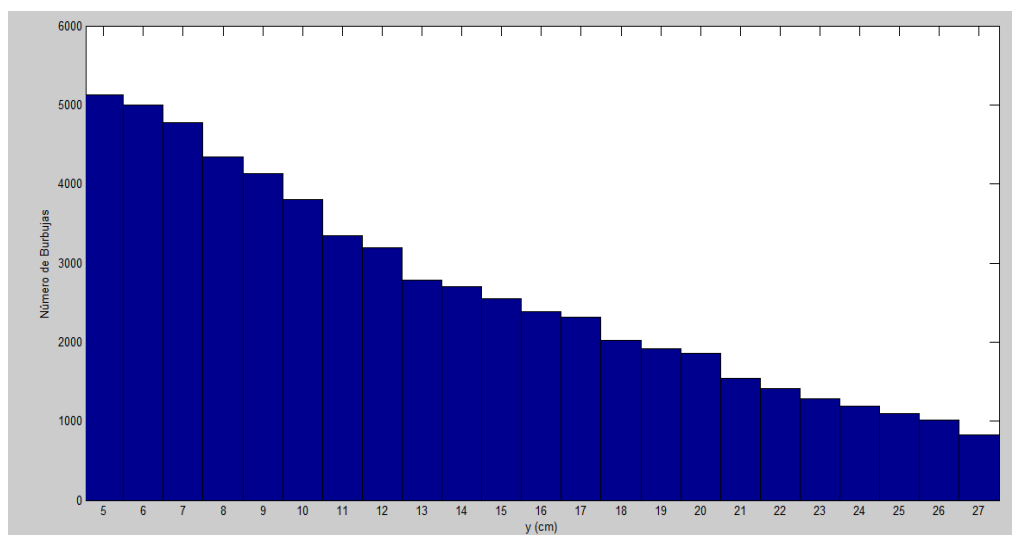


Fig 4.8. Número de burbujas observadas en función de la posición vertical en el lecho. Elaboración Propia

No se incluyen las figuras para el resto de sets de datos ya que no aportan nada a la exposición. Decir, eso sí, que se verificó mediante código incluido en el script de simulación que el número mínimo de burbujas por altura en cada set cumplía con las hipótesis estadísticas de muestra grande, incluso en el peor de los casos de estudio ($dy=0.5$ cm).

Para realizar el filtrado, utilizamos la función predefinida de Matlab ‘**grpstats**’ que utiliza la distribución t-Student para el cálculo del intervalo de confianza (se verificó accediendo al código fuente de la misma, accesible mediante el comando “open grpstats”), equivalente en el caso de muestra grande al intervalo de confianza con una distribución normal.

Tanto el espaciado de discretización para la altura en el que consideramos uniforme las propiedades de las burbujas, como el nivel de significación con el que realizamos los intervalos de confianza, son variables paramétricas del preproceso de los datos con los que alimentamos la red.

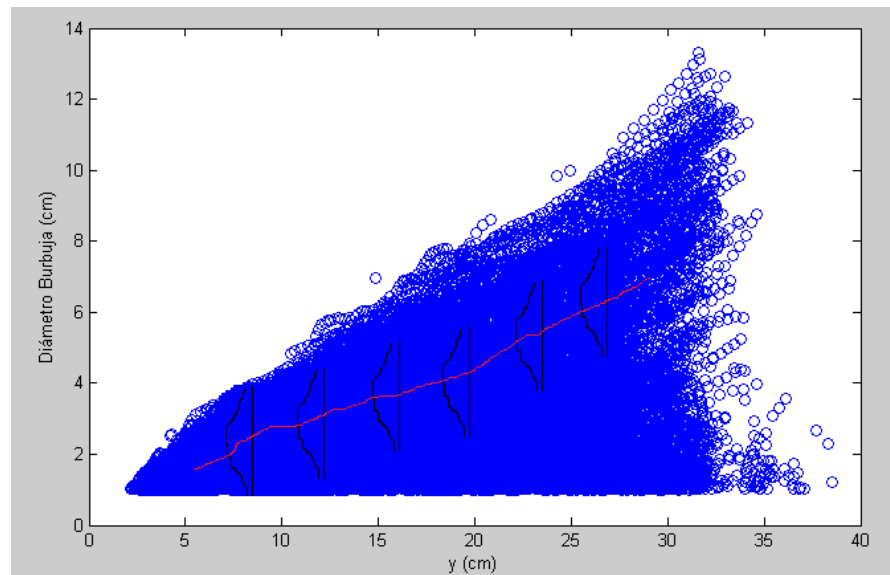


Fig 4.9. Gráfico explicativo del proceso de filtrado de datos por intervalos de confianza. Elaboración Propia

Para tratar de ilustrar la importancia de un correcto preprocesado de los datos, configuramos y entrenamos una red de 20 neuronas en la capa oculta sin realizar un filtrado previo de los datos de entrenamiento, obteniendo tras el proceso de aprendizaje para el set de datos con $h=30\text{cm}$ y $Q=94\text{ l/min}$ que la red no es capaz de ajustar los datos, tal y como muestra la salida del proceso de entrenamiento:

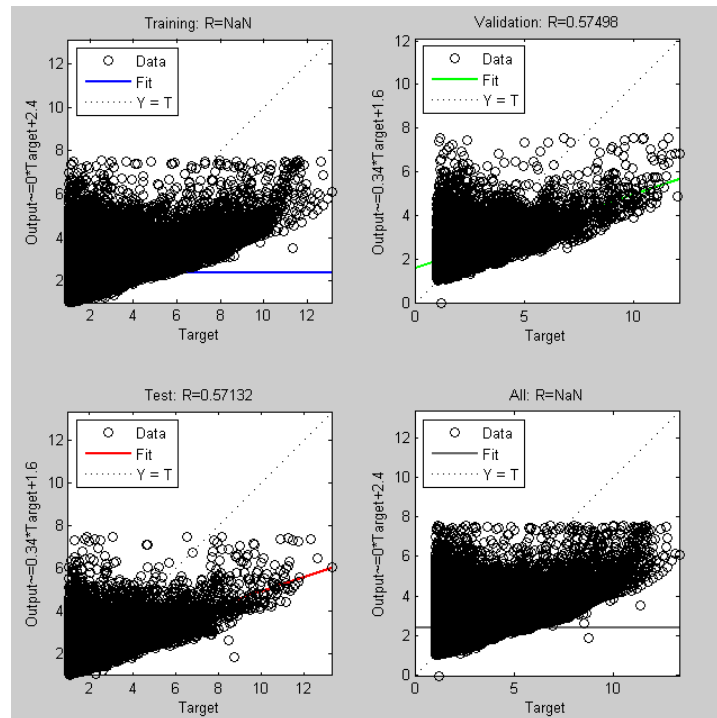


Fig 4.10. Resultados del entrenamiento con datos en bruto. Elaboración Propia

4.3 Variables para el diseño de la red y proceso de selección

Pasemos ahora a describir las variables que vamos a tener en cuenta para la optimización de la red:

- Variables del preproceso de los datos:
 - Se consideran diversos valores del espaciado de discretización “dy” de la altura, variando desde 0.5 a 2 cm.
 - También se estudia el comportamiento de la red en función del nivel de significación α con el que seleccionamos los datos dentro del intervalo de confianza para el diámetro medio en cada altura discretizada, variando entre 0.01 y 0.05
- Variables propias de la red neuronal:
 - Estructura de la red. Número de capas ocultas y de neuronas por capa.
 - Variables de entrada a la red: Incluir X_{cm} o no.
 - Utilizar una red común para diámetro y velocidad (capa de salida con 2 neuronas, o bien utilizar 2 redes por separado, una para el diámetro y otra para la velocidad).

Basándonos en el trabajo previo de María Izquierdo [Izq], realizado con datos obtenidos en el mismo lecho, pero con distintas alturas de lecho fijo y un preprocesado de datos único, decidimos tomar una red del tipo Perceptrón multicapa, '**newff**', con función de entrenamiento Levenberg-Marquardt, '**trainlm**', y función de activación '**tansig**' para restringir el rango de parámetros a tener en cuenta en el presente estudio, trabajando con el fin de mejorar los resultados previos. Además, estos parámetros coinciden con los que Matlab configura por defecto el perceptrón multicapa.

Nos decantamos por limitar el número máximo de pasos de entrenamiento, '**epochs**', permitidos a 100, y a utilizar '**validation stop**' para evitar el sobreentrenamiento y evitar la pérdida de capacidad de generalización. Estos criterios se basan en pruebas realizadas entrenando diversas redes, en la que observamos que el error en el entrenamiento apenas disminuye a partir de las 100 iteraciones, y que en ocasiones el error de validación comienza a aumentar, lo que hace recomendable parar el entrenamiento cuando durante seis pasos consecutivos la red haya aumentado el error para el conjunto de datos de validación.

El método empleado consiste en la programación de diversos scripts en Matlab (ver anexo) que recorren todo el rango de variación de los parámetros de diseño. Esto es, para cada valor de " **α** " y cada valor de "**dy**", creamos una red con diferente **número de capas y de neuronas ocultas en cada capa**, la entrenaremos y simularemos la red a través del conjunto de Test para ver la capacidad de generalización de la red. Además, este proceso se repite 10 veces para cada combinación de factores con el fin de reducir la variabilidad asociada a la inicialización de los pesos, e inferir con una determinada confianza estadística cuál será la mejor de las redes y los valores óptimos de los parámetros de filtrado.

4.4 Resultado de las simulaciones

4.4.1 Red con una capa oculta

Primeramente, probamos con una estructura como la que se muestra en la siguiente figura, variando el número de neuronas en la capa oculta desde 5 hasta 40, y el resto de parámetros de preprocesado de datos.

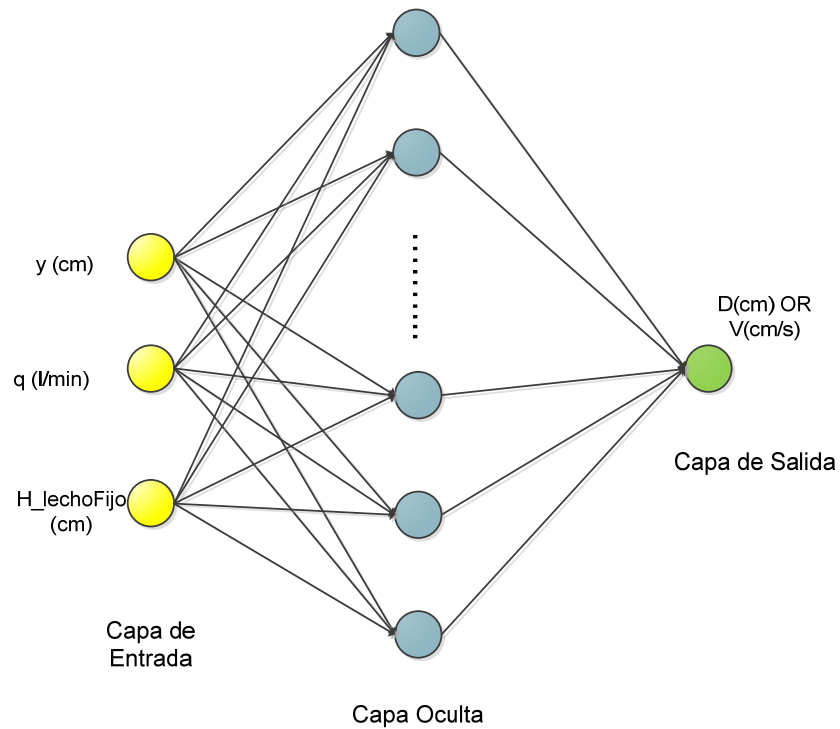


Fig 4.11. Estructura de la red con 1 capa oculta. Elaboración Propia

Red para simulación del diámetro

Tabla 4.3. Resultados predicción diámetro para Conjunto de Test. Red 1 capa oculta

dy	0.5	
alphaCI	0.01	
Promedio de RTest		
neuron	Total	
10	0.977	
15	0.979	
20	0.980	
25	0.981	

dy	0.5	
alphaCI	0.03	
Promedio de RTest		
neuron	Total	
10	0.981	
15	0.983	
20	0.984	
25	0.984	

dy	0.5	
alphaCI	0.05	
Promedio de RTest		
neuron	Total	
10	0.983	
15	0.985	
20	0.986	
25	0.987	

dy	1.25	
alphaCI	0.01	
Promedio de RTest		
neuron	Total	
10	0.987	
15	0.988	
20	0.990	
25	0.990	

dy	1.25	
alphaCI	0.03	
Promedio de RTest		
neuron	Total	
10	0.989	
15	0.990	
20	0.991	
25	0.992	

dy	1.25	
alphaCI	0.05	
Promedio de RTest		
neuron	Total	
10	0.989	
15	0.991	
20	0.992	
25	0.993	

dy	2	
alphaCI	0.01	
Promedio de RTest		
neuron	Total	
10	0.989	
15	0.991	
20	0.992	
25	0.992	

dy	2	
alphaCI	0.03	
Promedio de RTest		
neuron	Total	
10	0.990	
15	0.992	
20	0.993	
25	0.993	

dy	2	
alphaCI	0.05	
Promedio de RTest		
neuron	Total	
10	0.991	
15	0.993	
20	0.993	
25	0.994	

El 10% de los valores más altos del coeficiente de correlación aparecen marcados en rojo.

Si analizamos la variación entre tablas por columnas, vemos que un aumento en el tamaño de la discretización con la que filtramos los datos, dy, redundaba en una mejora del ajuste que proporciona la red. Seleccionamos entonces el valor de **dy=2 cm**.

Si nos movemos por filas, veremos cómo un aumento del nivel de significación con el que filtramos los datos en cada altura, igualmente parece mejorar el resultado de la red. Escojemos **alphaCI=0.05**.

Por último, en cada tabla, observamos como un aumento del número de neuronas también mejora los resultados. No obstante, ya que una disminución en su número proporcionaría una red con menos parámetros a entrenar, y por lo tanto más robusta y con menor coste computacional de entrenamiento, planteamos un contraste de hipótesis para ver si la diferencia entre ambos valores es significativa a nivel estadístico y poder elegir entre ellas con mejor criterio. El contraste empleado es el de igualdad de medias:

$$H_0 : \mu_1 - \mu_2 = 0$$

$$H_1 : \mu_1 - \mu_2 \neq 0$$

Introducimos los datos en el programa estadístico Minitab, que arroja los siguientes resultados:

Tabla 4.4. Salida del programa informático Minitab para un contraste de hipótesis

Two-Sample T-Test and CI: RTest, neuron					
Two-sample T for RTest					
neuron	N	Mean	StDev	SE Mean	
20	10	0.993169	0.000508	0.00016	
25	10	0.993691	0.000350	0.00011	
Difference = mu (20) - mu (25)					
Estimate for difference: -0.000522					
95% CI for difference: (-0.000938, -0.000107)					
T-Test of difference = 0 (vs not =): T-Value = -2.68 P-Value = 0.017 DF = 15					

Como el p-valor es pequeño (<0.05), concluimos que a un 95% de confianza SÍ existe una diferencia significativa en el rendimiento de ambas redes, por lo tanto elegimos la que contiene **25 neuronas** en su capa oculta.

Red para simulación de la velocidad

En este caso, la estructura de la red será la misma que en el caso anterior con la inclusión del diámetro de burbuja como entrada adicional.

Tabla 4.5. Resultados predicción velocidad para Conjunto de Test. Red 1 capa oculta

dy	0.5	↕
alphaCI	0.01	↕

Promedio de RTest		
neuron	Total	
10	0.944	
15	0.946	
20	0.947	
25	0.948	

dy	0.5	↕
alphaCI	0.03	↕

Promedio de RTest		
neuron	Total	
10	0.953	
15	0.955	
20	0.957	
25	0.956	

dy	0.5	↕
alphaCI	0.05	↕

Promedio de RTest		
neuron	Total	
10	0.958	
15	0.959	
20	0.960	
25	0.961	

dy	1.25	↕
alphaCI	0.01	↕

Promedio de RTest		
neuron	Total	
10	0.971	
15	0.973	
20	0.975	
25	0.975	

dy	1.25	↕
alphaCI	0.03	↕

Promedio de RTest		
neuron	Total	
10	0.975	
15	0.975	
20	0.978	
25	0.979	

dy	1.25	↕
alphaCI	0.05	↕

Promedio de RTest		
neuron	Total	
10	0.977	
15	0.978	
20	0.980	
25	0.981	

dy	2	↕
alphaCI	0.01	↕

Promedio de RTest		
neuron	Total	
10	0.978	
15	0.979	
20	0.981	
25	0.982	

dy	2	↕
alphaCI	0.03	↕

Promedio de RTest		
neuron	Total	
10	0.980	
15	0.982	
20	0.983	
25	0.984	

dy	2	↕
alphaCI	0.05	↕

Promedio de RTest		
neuron	Total	
10	0.981	
15	0.983	
20	0.984	
25	0.985	

En este caso observamos la misma tendencia a mejorar la predicción conforme se incrementa el valor de todas las variables consideradas.

Tomaremos entonces **dy=2cm, alphaCI=0.05**, y plantearemos otro contraste de hipótesis similar para elegir el número de neuronas en la capa oculta:

Tabla 4.6. Salida del programa informático Minitab para un contraste de hipótesis

Two-Sample T-Test and CI: RTest, neuron					
Two-sample T for RTest					
neuron	N	Mean	StDev	SE Mean	
20	10	0.98375	0.00122	0.00039	
25	10	0.98464	0.00153	0.00048	
Difference = mu (20) - mu (25)					
Estimate for difference:		-0.000892			
95% CI for difference:		(-0.002200, 0.000416)			
T-Test of difference = 0 (vs not =):		T-Value = -1.44		P-Value = 0.168	DF = 17

En este caso la diferencia entre utilizar 20 o 25 neuronas no es significativa a un nivel de confianza del 95%, por lo que optamos por la solución más sencilla, la red con **20 neuronas**.

4.4.2 Redes con dos capas ocultas

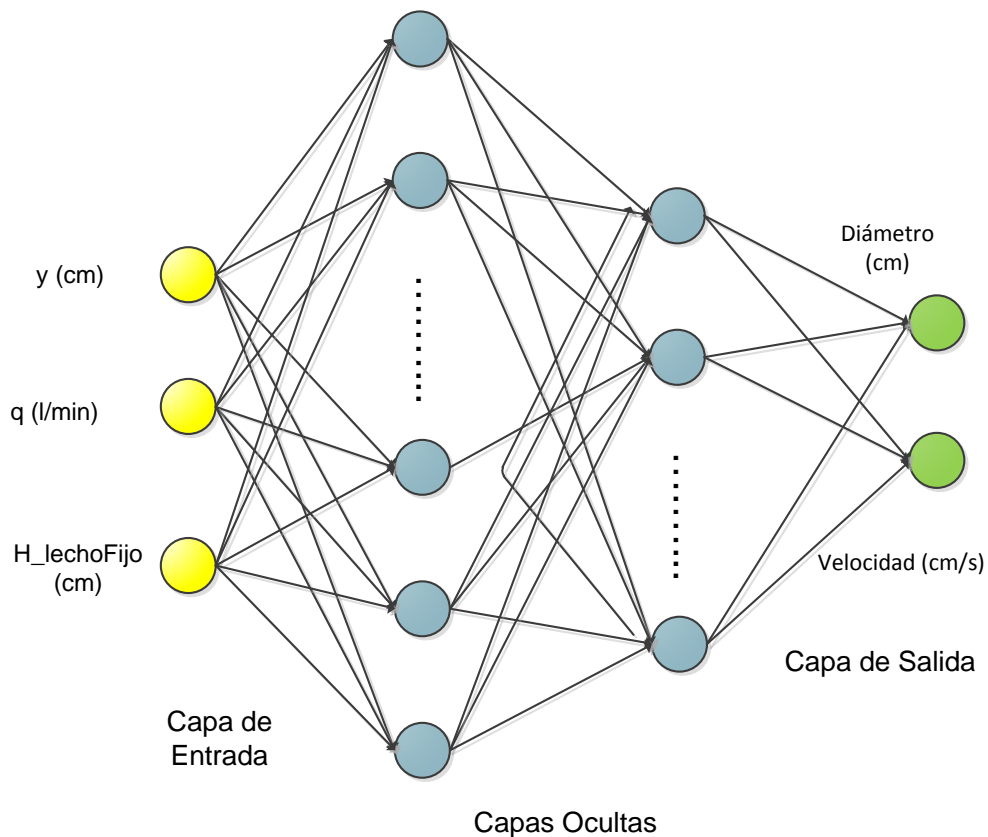


Fig 4.12. Estructura de la red con 2 capas ocultas. Elaboración Propia

Red para simulación del diámetro

Tabla 4.7. Resultados predicción diámetro para Conjunto de Test. Red 2 capas ocultas

dy	0.5	
alphaCl	0.01	
Promedio de RTest	neuron2	
neuron1	5	15
5	0.977	0.980
15	0.978	0.981

dy	0.5	
alphaCl	0.03	
Promedio de RTest	neuron2	
neuron1	5	15
5	0.982	0.984
15	0.984	0.986

dy	0.5	
alphaCl	0.05	
Promedio de RTest	neuron2	
neuron1	5	15
5	0.981	0.986
15	0.975	0.988

dy	1.25	
alphaCl	0.01	
Promedio de RTest	neuron2	
neuron1	5	15
5	0.988	0.989
15	0.986	0.992

dy	1.25	
alphaCl	0.03	
Promedio de RTest	neuron2	
neuron1	5	15
5	0.987	0.991
15	0.953	0.993

dy	1.25	
alphaCl	0.05	
Promedio de RTest	neuron2	
neuron1	5	15
5	0.988	0.992
15	0.983	0.994

dy	2	
alphaCl	0.01	
Promedio de RTest	neuron2	
neuron1	5	15
5	0.985	0.992
15	0.991	0.994

dy	2	
alphaCl	0.03	
Promedio de RTest	neuron2	
neuron1	5	15
5	0.990	0.993
15	0.992	0.994

dy	2	
alphaCl	0.05	
Promedio de RTest	neuron2	
neuron1	5	15
5	0.976	0.994
15	0.991	0.994

Pese a que las tendencias se mantienen (aumentando el espaciado de discretización de la altura, el nivel de significación con el que filtramos los datos, e incluyendo un mayor número de neuronas, se obtiene una disminución del error), observamos que se presentan ciertos valores más bajos de lo esperado, remarcados en amarillo.

Si entramos al detalle de las redes individuales que generan estos promedios que no siguen la tendencia, veremos que es debido a que en ocasiones no se entrenan correctamente, mientras que en otras lo hacen con normalidad. Esto ocurre porque durante el entrenamiento, el error en el conjunto de validación comienza a aumentar, con lo cuál se detiene el entrenamiento.

Tabla 4.8. Resumen de los diferentes entrenamientos de una estructura de red con resultados atípicos.

neuron1	neuron2	dy	alphaCl	tr.best_epoch	max(tr.time)	mTest	bTest	RTest
15	5	1.25	0.03	56	19.905	0.99627373	0.010643817	0.887099417
15	5	1.25	0.03	100	33.593	0.998617016	0.005170617	0.99236144
15	5	1.25	0.03	100	32.264	1.003017836	-0.016949851	0.992658928
15	5	1.25	0.03	100	32.226	1.000841348	-0.003068523	0.992138722
15	5	1.25	0.03	99	32.794	1.003069484	-0.010585056	0.991808974
15	5	1.25	0.03	99	31.855	1.004658427	-0.016649275	0.992760057
15	5	1.25	0.03	100	31.964	1.000969485	-0.001709399	0.99280862
15	5	1.25	0.03	68	23.587	0.987422887	0.037894799	0.883825056
15	5	1.25	0.03	71	24.913	0.988952654	0.051906341	0.809982158
15	5	1.25	0.03	100	31.7	0.999890465	0.007905595	0.992208543

Ya que los valores del coeficiente de correlación para los datos de Test son, en el mejor de los casos equivalentes a los que obteníamos con redes de una capa ($R=0.994$), nos quedaremos siempre la estructura más simple, que además evita que se produzcan estos problemas derivados de tener un mayor número de parámetros a configurar, con lo que nuestra red será más robusta.

Cabe destacar igualmente que se producen cuando el número de neuronas en la segunda capa oculta es bajo, y aún más si en la primera capa es elevado. Esto se ha comprobado con varias simulaciones realizadas, considerando diferentes combinaciones de número de neuronas por capa.

Red para simulación de la velocidad

Tabla 4.9. Resultados predicción diámetro para Conjunto de Test. Red 2 capas ocultas

dy	0.5	
alphaCl	0.01	

Promedio de RTest	neuron2	
neuron1	5	15
5	0.944	0.949
15	0.950	0.952

dy	0.5	
alphaCl	0.03	

Promedio de RTest	neuron2	
neuron1	5	15
5	0.952	0.957
15	0.904	0.960

dy	0.5	
alphaCl	0.05	

Promedio de RTest	neuron2	
neuron1	5	15
5	0.933	0.962
15	0.961	0.965

dy	1.25	
alphaCl	0.01	

Promedio de RTest	neuron2	
neuron1	5	15
5	0.971	0.976
15	0.977	0.977

dy	1.25	
alphaCl	0.03	

Promedio de RTest	neuron2	
neuron1	5	15
5	0.975	0.979
15	0.977	0.981

dy	1.25	
alphaCl	0.05	

Promedio de RTest	neuron2	
neuron1	5	15
5	0.975	0.981
15	0.982	0.984

dy	2	
alphaCl	0.01	

Promedio de RTest	neuron2	
neuron1	5	15
5	0.978	0.982
15	0.982	0.985

dy	2	
alphaCl	0.03	

Promedio de RTest	neuron2	
neuron1	5	15
5	0.980	0.984
15	0.982	0.988

dy	2	
alphaCl	0.05	

Promedio de RTest	neuron2	
neuron1	5	15
5	0.946	0.986
15	0.987	0.988

Como puede observarse, el patrón es muy similar al del resto de configuraciones ya vistas. Al igual que en el caso anterior, las redes de dos capas con bajo número de neuronas en la segunda se comportan ocasionalmente de manera atípica durante el entrenamiento, tal y como se ve en la siguiente tabla.

Para este caso, una estructura de red [5][15] o [15][15] ofrece mejores resultados que una red monocapa. Esto debería ser comprobado mediante un contraste de hipótesis, pero ya que la diferencia es pequeña (0.9888 frente a 0.9885), selecciono directamente la red de 1 capa, quedándome en el lado de la seguridad.

Tabla 4.10. Resumen de los diferentes entrenamientos de una estructura de red con resultados atípicos.

neuron1	neuron2	dy	alphaCI	tr.best_epoch	max(tr.time)	mTest	bTest	RTest
15	5	0.5	0.03	35	29.424	1.001289557	-0.021508053	0.949629654
15	5	0.5	0.03	83	63.12	0.997552122	0.062635513	0.960022949
15	5	0.5	0.03	79	69.183	0.996866024	0.077181792	0.958413602
15	5	0.5	0.03	100	71.171	1.002177586	-0.028599825	0.959036971
15	5	0.5	0.03	99	72.545	0.991133381	0.257088481	0.955571559
15	5	0.5	0.03	12	8.611	0.988200233	0.384656307	0.423377848
15	5	0.5	0.03	99	68.126	0.99576626	0.123301528	0.956945154
15	5	0.5	0.03	81	59.343	1.001428187	-0.010466587	0.961451065
15	5	0.5	0.03	100	68.188	0.995007356	0.140438989	0.959401579
15	5	0.5	0.03	99	67.86	1.000449306	0.004134504	0.954039012

4.4.3 Red conjunta para diámetro y velocidad (dos neuronas de salida, monocapa)

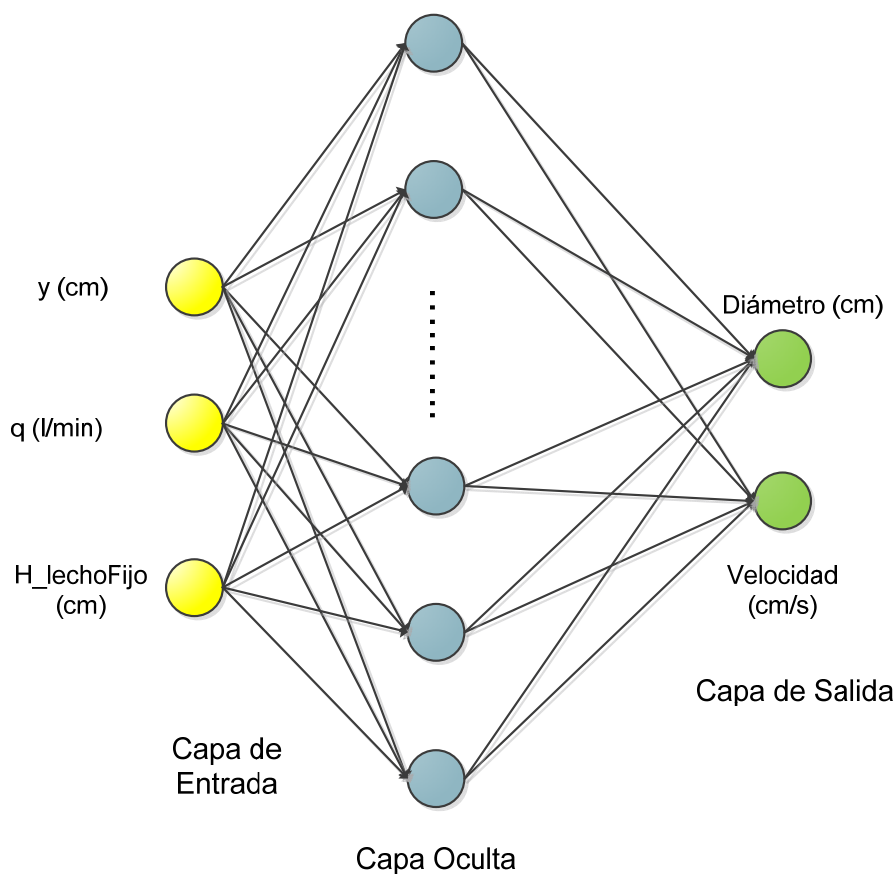


Fig 4.13. Estructura de la red con 1 capa oculta y 2 neuronas de salida. Elaboración Propia

Tabla 4.11. Resultados predicción diámetro y velocidad para Conjunto de Test. Red 2 capas ocultas y 2 neuronas en la capa de salida

dy	0.5		dy	0.5		dy	0.5	
alphaCl	0.01		alphaCl	0.03		alphaCl	0.05	
Datos			Datos			Datos		
neuron	Promedio de RTestD	Promedio de RTestVel	neuron	Promedio de RTestD	Promedio de RTestVel	neuron	Promedio de RTestD	Promedio de RTestVel
10	0.938	0.897	10	0.942	0.918	10	0.946	0.920
15	0.946	0.904	15	0.951	0.920	15	0.956	0.926
20	0.944	0.903	20	0.954	0.919	20	0.963	0.926
25	0.950	0.908	25	0.959	0.924	25	0.961	0.930

dy	1.25		dy	1.25		dy	1.25	
alphaCl	0.01		alphaCl	0.03		alphaCl	0.05	
Datos			Datos			Datos		
neuron	Promedio de RTestD	Promedio de RTestVel	neuron	Promedio de RTestD	Promedio de RTestVel	neuron	Promedio de RTestD	Promedio de RTestVel
10	0.956	0.942	10	0.961	0.953	10	0.960	0.956
15	0.963	0.948	15	0.966	0.959	15	0.966	0.961
20	0.964	0.952	20	0.975	0.960	20	0.972	0.960
25	0.968	0.953	25	0.971	0.950	25	0.972	0.959

dy	2		dy	2		dy	2	
alphaCl	0.01		alphaCl	0.03		alphaCl	0.05	
Datos			Datos			Datos		
neuron	Promedio de RTestD	Promedio de RTestVel	neuron	Promedio de RTestD	Promedio de RTestVel	neuron	Promedio de RTestD	Promedio de RTestVel
10	0.959	0.965	10	0.960	0.963	10	0.963	0.965
15	0.970	0.967	15	0.968	0.968	15	0.967	0.966
20	0.975	0.968	20	0.973	0.971	20	0.971	0.966
25	0.976	0.969	25	0.977	0.969	25	0.977	0.968

Puede observarse que en este caso los resultados empeoran notablemente con respecto al caso en el que simulabamos por separado diámetro y velocidad.

Esto se debe principalmente al hecho de la reducción drástica del número de datos de entrenamiento, y no tanto a que la red tenga dos neuronas de salida, ya que en este caso el filtrado de los datos que caen dentro del intervalo de confianza para la media se realiza para el diámetro y para la velocidad, con lo cuál el número de burbujas que cumplen ambas condiciones simultáneamente es mucho menor (digamos que las redes con una salida son entrenadas con cerca de 20.000 patrones, mientras que las de dos salidas con unos 1.000, es decir, 20 veces menos).

4.4.4 Inclusión de Xcm como variable de entrada

Teóricamente, no debería influir la posición en x de la burbuja, salvo cuando se encuentren cerca de la pared y noten su influencia, casos que ya eliminamos antes de entrenar a las redes.

No obstante, probamos a simular por si hubiera algún efecto que no estuviéramos considerando, obteniendo, para la misma estructura de red, valores ligeramente inferiores ($R=0.993$ frente a $R=0.994$ que obtuvimos sin considerar x), con lo que concluimos que es mejor no considerarla como variable de entrada a la red.

4.5 Conclusiones y resumen del capítulo

A modo de resumen:

- Para simular el diámetro, seleccionamos una red con una única capa oculta de 25 neuronas, con los siguientes parámetros de preproceso de los datos que la alimentan: Espaciado de discretización vertical $dy=2\text{cm}$; nivel de significación del intervalo de confianza para el filtrado, $\alpha_{CI}=0.05$. La medida promedio del coeficiente de correlación de la salida simulada por la red con la obtenida experimentalmente es de $R=0.994$ para el conjunto de Test.
- Para simular la velocidad, seleccionamos una red con una única capa oculta de 20 neuronas por capa y mismos parámetros de procesamiento de datos que en el caso anterior. La medida promedio del coeficiente de correlación de la salida simulada por la red con la obtenida experimentalmente es de $R=0.984$ para el conjunto de Test.
- En general, las redes de una capa oculta son más robustas que las de dos para este problema, especialmente si el número de neuronas en la segunda capa oculta es bajo, por lo que a pequeñas diferencias de rendimiento medio, elegimos siempre las más sencillas y estables.
- En este problema, dado el método de preprocesado de los datos propuesto, es mejor crear redes separadas para predecir el diámetro y la velocidad, que una sola red con dos neuronas en la capa de salida. Esto nos permite incluir también el diámetro como variable de entrada para la red de simulación de la velocidad, e incluir un mayor número de datos para el entrenamiento ($\times 20$).
- No es conveniente incluir la posición horizontal del centro de masas de la partícula como variable de entrada a la red, ya que aumenta el tiempo de procesamiento y no aporta mejora alguna.
- El análisis efectuado de la sensibilidad de la red a los datos con los que se la alimenta, muestra la importancia del preproceso de los mismos antes del entrenamiento. El método seguido en el presente proyecto, en el que se optimiza la capacidad de generalización de la red dependiendo de esta etapa de filtrado previa, puede servir de guía para trabajos futuros.

Trabajo a futuro

- Incluir como variable de salida a predecir por la red el número medio de burbujas que hay en cada altura.
- Mejorar el trabajo presente mediante un análisis que investigue los datos fuera de rango que proporciona la salida del programa de reconocimiento de imágenes (que son los datos de partida para el presente proyecto), para determinar los límites reales de tamaño máximo de burbuja y velocidad máxima.
- Obtener datos experimentales para combinaciones intermedias de caudales y alturas de lecho fijo, que nos sirvan de Test para evaluar la capacidad que tiene la red para “interpolarse” o predecir los valores de diámetro y velocidad con datos de diferente caudal y diferente altura respecto a los usados en su entrenamiento.

Referencias Bibliográficas

[KL] Kunni, D., Levenspiel, O.: 'Fluidization Engineering' (Butterworth-Heinemann, 2nd ed., 1991).

[Izq] Izquierdo, M.: 'Caracterización de lechos fluidos Burbujearantes'. Proyecto fin de Master, Universidad de Vigo.

[YAN] 'Handbook of fluidization and fluid particle systems', (Ed Wen-Ching-Yan, 2003).

[MS] Martín, B., Sanz, A.: 'Redes Neuronales y Sistemas Borrosos' (Ra-Ma, 3th ed, 2006).

[SSM] Serrano, A., Soria, E., Martín, J.: 'Redes neuronales artificiales' (Universidad de Valencia, OpenCourseWare). Disponible [Internet]: http://ocw.uv.es/ingenieria-y-arquitectura/1-2/libro_ocw_libro_de_redes.pdf

[MAT] Matlab R2009a, 'Users Guide, Neural Network Toolbox'

[MSF] Martín, I., Salcedo, R., Font, R.: Mecánica de Fluidos, Tema4. 'Transporte cantidad movimiento sólido-fluido. Disponible [Internet]: http://rua.ua.es/dspace/bitstream/10045/20299/9/tema4_flujo%20externo.pdf

[RN] Ruud van Ommen, J., Naoko Ellis. 'Fluidization', JMBC/OSPT course Particle Technology. Disponible [Internet]: http://www2.msm.ctw.utwente.nl/sliding/TEACHING/ParticleTechnology/vanOmmen_Fluidization.pdf

[UHU] 'Combustión en lecho fluidizado'. Disponible [Internet]: http://www.uhu.es/prochem/wiki/index.php/Combusti%C3%B3n_en_lecho_fluidizado

[NEU] <http://www.uhu.es/francisco.cordoba/asignaturas/FBAM/TEMAS%20PDF/3-LA%20NEURONA.pdf>

[CS] http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php

[ISE] <http://ise.uc3m.es/>